REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Services and Communications Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	01/19/10
2. REPORT TYPE	Final Report
3. DATES COVERED *(From - To)*	November 2007 – December 2009
4. TITLE AND SUBTITLE	DARPA/IPTO Final Report Option 1/Phase III
5a. CONTRACT NUMBER	HR0011-06-C-0022
5b. GRANT NUMBER	
5c. PROGRAM ELEMENT NUMBER	CLIN 000603
5d. PROJECT NUMBER	
5e. TASK NUMBER	
5f. WORK UNIT NUMBER	
6. AUTHOR(S)	Dr. John Makhoul
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	BBN Technologies 10 Moulton Street Cambridge MA 02138
8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	Defense Advanced Research Projects Agency DARPA/IPTO 3701 North Fairfax Drive Arlington VA 22203 Dr. Joseph Olive
10. SPONSOR/MONITOR'S ACRONYM(S)	
11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT	TBD
13. SUPPLEMENTARY NOTES	
14. ABSTRACT	Final Report
15. SUBJECT TERMS	

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Diane Messuri
U	U	U			19b. TELEPHONE NUMBER *(Include area code)* 617-873-2449

AGILE: Autonomous Global Integrated Language Exploitation

Final Report (Year 3)

Contractor: **BBN Technologies**
10 Moulton Street
Cambridge, MA 02138

Principal Investigator: **Dr. John Makhoul**
Tel: 617-873-3332
Fax: 617-873-2473
Email: makhoul@bbn.com

Reporting Period: **November 2007 – December 2009**

This material is based upon work supported by the
Defense Advanced Research Projects Agency DARPA/IPTO
AGILE: Autonomous Global Integrated Language Exploitation
ARPA Order No.: V002
Program Code No.: 5M30
Issued by DARPA/CMO under Contract #HR0011-06-C-0022

20100122008

DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

DTIC® has determined on 1/26/2010 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited. per DARPA

☐ **© COPYRIGHTED**; U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

TABLE OF CONTENTS

1 Introduction

This is the final report for Year 3 of the GALE project, whose objective is to transcribe and translate foreign spoken and written languages into English and to distil the transcription into accurate information for use by our military. Below, we summarize the work performed by the BBN-led AGILE Team in Year 3. A more detailed description of the work performed can be found in the DARPA/IPTO Quarterly Status Reports for this project.

The Appendix contains the accomplishments of three additional efforts: Serif Maturation, Broadcast Monitoring System One-Year Archive, Robust Automatic Transcription of Speech (RATS) and Serif Research.

2 GALE Program Go/No-Go Targets

In the Phase 3 evaluations, the AGILE team passed the Arabic Phase 3 Go/No-Go targets for all four conditions: Newswire (NW), Web text (WB), Broadcast News (BN), and Broadcast Conversations (BC). For the Chinese Phase 3 evaluation, the team passed the target for WB and came close in the three other conditions. The results for both Arabic and Chinese were the best among the participating teams.

After the evaluations, we performed extensive error analysis on documents that did not pass the targets, compiled a list of phenomena that are causing the majority of errors, and proposed methods to deal with these phenomena in Phase 4 of the program. One conclusion from the analysis was that Chinese was significantly more difficult to translate into English than Arabic. That additional difficulty for translating Chinese was also borne out in experiments with human translators, where the human Chinese translations had lower accuracy than Arabic translations. Our conclusion was that, in Phase 4, we need to put special effort in solving Chinese-related translation problems.

In the Distillation evaluations, BBN passed all the Go/No-Go targets for Phase 3.

3 Accomplishments in Speech-to-Text (STT)

3.1 BBN Technologies

During Phase 3 of the GALE Program, we made significant progress in our Speech-To-Text (STT) systems at BBN, achieving 12-17% relative reduction in word error rate (WER) for Arabic and 5-9% relative reduction in character error rate (CER) for Mandarin. Details of STT performance improvement as measured on the development sets of Phase 2 and Phase 3 (dev07 and dev08, respectively) as well as the evaluation set of Phase 3 (eval08) are captured in Table 3-1 and Table 3-2 below. Major contributions to this improvement include the use of manual audio segmentation, better technique to pre-process acoustic training data, better procedure to make up phonetic pronunciations for Arabic words, the use of additional training data, the creation of a new morpheme-based Arabic STT system using morphemes derived from a contextual morphological analysis, and a new procedure to compound Chinese words based on Chinese parse trees. In

addition to the average error rates, the tables also provide (between parentheses) the error rates for Broadcast News (BN) and Broadcast Conversations (BC).

System	dev07 (BN/BC)	dev08 (BN/BC)	eval08 (BN/BC)
P2	11.0 (9.1/14.5)	13.0 (9.4/18.6)	------
P3	9.7 (7.8/13.1)	10.8 (7.9/15.5)	9.5 (6.8/12.3)

Table 3-1: Comparison of WERs produced by BBN's Phase 2 (P2) and Phase 3 (P3) Arabic STT systems.

System	dev07 (BN/BC)	dev08	eval08
P2.5	9.4 (2.7/14.5)	8.8 (3.4/14.1)	12.3 (4.9/19.6)
P3.5	8.9 (2.5/13.7)	8.3 (3.0/13.5)	11.7 (4.6/18.7)

Table 3-2: Comparison of CERs produced by BBN's Phase 2 (P2.5) and Phase 3 (P3.5) Mandarin STT systems.

By incorporating manual audio segmentation into our decoding systems, we observed a 6% relative WER reduction for Arabic and 5% relative reduction in CER for Mandarin.

Arabic STT

We improved the acoustic models (AM) by reprocessing all of the available acoustic training data through light supervision using a graphemic system to make sure no data was excluded because of out-of-vocabulary words (due to lack of phonetic pronunciations). Arabic dialect affixes were added to the list of affixes used in the Buckwalter morphological analyzer so that most dialect words could be vocalized to provide data to derive their phonetic pronunciations automatically. Pronunciations from the manually-vocalized corpora provided by LDC were also used to upgrade our Arabic 1.2M-word master phonetic dictionary. These efforts resulted in 0.3-0.7% absolute WER reduction for the phonetic system.

The Arabic language models (LM) were also improved by including the additional texts from Cambridge University and LDC, and by using a larger held-out set in the LM interpolation. We also revised our Arabic recognition vocabularies by increasing their sizes as well as updating the list of frequent Arabic words. The better LM provided 0.1-0.3% absolute reduction in WER.

In addition to the three existing systems (phonetic, graphemic, and simple morphemic) as used in Phase 2, we developed a new morpheme-based system using morphemes derived from a morphological analysis provided by Sakhr Software, which uses context to provide unique morphological tags. This latest morpheme-based system is currently the best of our four Arabic STT systems. With this additional system, the ROVER combination of four systems produced better WER than the combination of three systems.

Mandarin STT

In addition to the improvement obtained by using manual audio segmentation and extra training data, we obtained substantial gain by developing a new procedure to compound Chinese words based on Chinese parse trees. We parsed all of our LM training data and collected statistics of the sequences of sibling words (i.e. sequences of leaves from the same ancestor). Each of the 40K most frequent sequences of sibling words was designated as a "compound" Chinese word. This word compounding scheme provided 0.1-0.2% absolute CER reduction.

Improved Translation of Arabic speech

We investigated a few methods to improve translation of Arabic speech. First, we found that the audio parallel data is especially useful for translation of Arabic speech, although the amount of data is very small. Second, we added STT hypotheses to the parallel training corpus by pairing them with the reference translations, which gave about 0.4 to 0.8 improvement in TER-BLEU score, computed as (TER-BLEU)/2.

Optimizing STT System Combination for Machine Translation (MT)

We continued to investigate MT-based tuning of the STT system combination parameters, and so far modest gains have been achieved by combining several individual STT system outputs with weights tuned to minimize a modified word error rate (WER) function that assigns a larger cost to deletion errors.

3.2 Cambridge University

Improved Arabic STT System

An improved Cambridge HTK-based Arabic STT system was produced for the AGILE Phase 3 Arabic evaluation. Major new features included building systems with new data; improved language models including the use of automatically generated class-based models; work on pronunciation, generation; and the inclusion of multi-layer perceptron features in the observation vector of the acoustic models. Taken together these methods produced an evaluation system with a word error rate 16% relative lower than the Cambridge system for the GALE Phase 2 evaluation. The system includes both graphemic and phonetic acoustic model branches and cross-adaptation between branch types.

MLP-based Feature Extraction

The use of multi-layer perceptron (MLP) based front end feature extraction has been investigated and developed. We have used both PLP and a TRAP-based acoustic analysis as input to these models. The MLP is trained to predict phone posterior probabilities and a bottleneck layer is used to extract the required features. Typically 26 such features will be appended to the standard PLP-based HMM feature vector. Large reductions in error rate are observed with maximum likelihood training. When using discriminative training, unsupervised adaptation and system combination are used, the improvements with MLP features are much reduced but still worthwhile and such systems were used for both the Arabic and Chinese Cambridge STT systems. One item of note is that we have found that using MLP systems trained with phonetic targets can produce features of great benefit to Arabic graphemic systems.

Improved Phonetic Models for Arabic STT

We have investigated methods of deriving sets of pronunciations for unknown words which allows the lexical coverage of a phonetic Arabic STT system. Both rule-based approaches using training data phonmetic alignments and graphone-based models have been investigated.

Morphological Decomposition using MADA

Due to the productive morphology of Arabic, there are a very large number of word forms observed in Arabic texts. Hence the Arabic language models are very sparse and OOV rates tend to be high. We have investigated a morphological decomposition technique using the MADA toolkit which performs both decomposition and root normalization and complete Arabic STT systems have been built using MADA-based processing. To obtain a word-level output it is necessary to translate back to the word domain, and this is achieved using a statistical machine translation approach. This technique has led to reductions in error rates of up to 1% absolute. It also has led to interesting direct integration with MT systems using MADA-based STT lattices and small improvements in translation accuracy.

Class Based and Sub-Word Language Models for Arabic STT

Automatically derived class-based language models have been investigated for Arabic for vocabularies up to 350k words and have led to reductions in error rates.

Improved Chinese STT System

The performance of the Cambridge Chinese STT system was improved by about 9% relative through the use of MLP features and context-dependent language model adaptation. The AGILE Chinese STT system used outputs from both the BBN and the LIMSI systems for adaptation supervision for the Cambridge system which generates the final output. This form of combination was optimized for translation performance but also reduced speech recognition error rates by about 9% relative over the year.

Multilevel Chinese Language Models

The use of character-level language models in combination with the usual word-based models has been investigated. If log-linear combination between lattices, rather than use of e.g. ROVER combination, then small improvements in character error rate result.

Advanced Acoustic Modeling

A number of advanced techniques for acoustic modeling have been investigated. These include a method of discriminative estimation of adaptation transforms known as discriminative mapping transforms (DMT); the use of DMT in adaptive training and Bayesian adaptive training. Other work has focused on improved discriminative training criteria. Work is continuing in these areas.

Context-Dependent Language Model Adaptation and Cross-Adaptation

The N-gram language models for STT are an interpolation of a number of component language models from a number of sources/genres. Unsupervised language model adaptation is performed so as to tune the language model for a particular story by varying the interpolation weights based on an initial transcription with a non-adaptive system.

This interpolation can be more effective by using context-dependent interpolation weights for different word histories. These weights are either adapted to minimize perplexity or to minimize the expected STT error rate. This method was key in the AGILE Chinese STT system where language model cross-adaptation was used.

Data and Software Released

Cambridge released a new version of the HTK toolkit HTK V3.4.1 which includes a number of bug fixes, improved lattice generation for the HTK large voculary decoder, improved documentation for discriminative training and the large vocabulary decoder and a number of other enhancements of functionality. Various web-collected data sets were released to the GALE community for language modeling in Arabic and Chinese.

3.3 LIMSI

The main LIMSI activities in the third year of the GALE program have addressed improving speech-to-text for Arabic and Mandarin. LIMSI provided component systems for the AGILE participation in the Year 3 evaluations. Most of our work has been a continuation of work started earlier in the program, in particular developing methods to train with incomplete information for Arabic; the inclusion of prosodic features for Mandarin; the exploration of discriminative long span features estimated with multi-layer perceptrons and how to fuse them with traditional cepstral features; a revised language model training procedure; updated neural network language models and revised decoding strategies. Overall the word error rate of the Arabic system has been reduced by 20% relative to the year 2 (June 2007) system for all development test sets, and the Mandarin character error rate was reduced by over 15% relative to the previous LIMSI system.

Work has continued exploring Multi-Layer Perceptron (MLP) front-ends with four sets of raw features as well as their combination with cepstral ones, both without and with unsupervised model adaptation. These features augment the short-term spectral representation provided by PLP or MFCC features with more contextual information. The raw features initially explored were nine frames of PLP (9xPLP) and warped linear predictive temporal patterns (wLP). These features are costly to calculate since they use very large FFT transformations. To get around with this handicap, two other types of raw features were investigated, Multi-RASTA (MR) and TRAP-DCT (TD). The wLP and TRAP-DCT (TD) features were found to have comparable performances both standalone and when concatenated with traditional PLP features, and also were found to be more complementary to PLP features than the 9xPLP and multi-resolution RASTA based features. This is very interesting since raw TD features are only slightly more costly to compute than PLP features, and much less costly than the wLP ones. For the Arabic language, all the MLPs and the MLP-HMM models were trained on 1200 hours of manually transcribed data. SAT training, discriminative training and acoustic model adaptation have been successfully used with these models. While more extensive investigations were made for the Arabic language, similar observations were made for Mandarin and full systems have been built. Using concatenated PLP and MLP features in a full system (SAT, MMI training, multi-pass decoding with unsupervised adaptation) gives a relative error reduction of on the order of 5% compared to a fully trained PLP system, and ROVER combination can also given an additional gain.

A method for morphological decomposition in Arabic was developed, derived from the Buckwalter morphological analysis. Rules were added to block decomposition under certain conditions (if there are multiple possible decompositions or if the stem contains fewer than three letters) as well as a constraints to not decompose the 65k most frequent words and to block the decomposition of the prefix 'Al' if the stem begins with a solar consonant. This method has been evaluated using the complete set of available audio and textual data. The performance of the fully-trained system using morphological decomposition is comparable to that of the fully trained MLE trained word-based system. Combining the word based and morph based systems using Rover gives a word error reduction across all GALE development and test sets of about 0.6%.

A generic vowel was introduced to facilitate training with unvocalized Arabic transcripts. The acoustic models used in the LIMSI GALE systems were built using dictionaries for which about 15% of the words have generic vowels in their pronunciation. Simplified generic vowel rules in which a generic vowel is inserted after each consonant and semivowels are mapped to their corresponding long vowels were developed to enable these words to be included in the recognition lexicon. In initial tests adding words with generic vowels to the recognition lexicon, can recover about one-third of the errors due to these previously out-of-vocabulary words.

Different pitch algorithms and smoothing techniques (linear interpolation, CU mode interpolation) with the ESPS pitch features were tested on four Mandarin development sets. The best smoothing method was found to be a linear interpolation between voiced segments. (A Piecewise Cubic Hermite Interpolating Polynomial smoothing was found to degrade the CER by about 0.5%). The incorporation of pitch features in the Mandarin system gave an absolute CER reduction of about 1-3% depending upon the data set with a single decoding pass, and about 0.5% with a full system (with multi-pass decoding and adaptation).

The LM tools and training process were completely redesigned in order to build language models for Mandarin evaluation system. This allowed much larger models to be built than in the past and to exhaustively investigate the impact of various discounting and interpolation techniques on the model accuracy. Large n-gram language models are generally formed by interpolating component LMs estimated on a number of text subsets, where the interpolation coefficients optimized on development data. Different recipes are used in which cutoffs or pruning can be applied during the process. One of the main conclusions from the experiments is that it is best to avoid applying cutoff thresholds or entropy pruning on the individual models before the final interpolation, The best strategy is do all pruning at the end, so as to not discard any information up to the end of the training process, where the model size can be optionally (and optimally) reduced with entropy pruning. For example, to have comparable perplexities with a 3-gram LM, pruning before merging results in an LM that is three times as large as the one obtained by merging first, and for comparable size 4-gram LMs, the perplexity with a model obtained by pruning before interpolation is 30% higher than when pruning after interpolation.

Since much of the algorithmic and methodology advances are language independent, the methods that have been developed and found to be successful for one of the languages are ported to the STT system for the other language. For example, the new language

model training method and pitch features are being incorporated in the Arabic system. More extensive experiments were carried out for the Arabic language, with the best configurations then applied in the Mandarin STT system.

4 Accomplishments in Machine Translation (MT)

4.1 BBN Technologies

Data Processing for MT

During the development of the BBN P3.5 Chinese-English MT system, we devoted some effort to improve the normalization/tokenization of training and testing data. We found that many word alignment problems were due to poor matching or inconsistent representation of numbers on the two sides of the training bi-text, as well as due to incorrect character-to-word segmentation of the Chinese source text. Small, but consistent MT improvements were obtained by addressing these two issues. Some additional small gain was also achieved by automatically extracting Chinese-English name pairs from the parallel training data and adding them to the training with a high weight.

Error Analysis

We performed extensive error analysis on a subset of the P3 Arabic evaluation set. Our results showed that various types of words are frequently dropped in the translation. This indicates the need to focus future work in feature development, parameter tuning architecture, etc., and on minimizing deleted terms, as well as on ensuring the proper insertion of items like subject pronouns or copular verbs, which are implicit in Arabic. The results also indicated a significant role for sentence structure improvements, such as improved use of source/target syntax, lexicalized or structure-based reordering models, and features based on literal semantics, such as propositions.

Corpus Weight Estimation

We revised the implementation of our corpus weight estimation technique, thus allowing experimentation with sentence-level weights. In the new scheme, each sentence in the training bi-text can receive a weight that is used to influence the calculation of the probabilities for all rules extracted from the bi-text, and the weights are estimated discriminatively to minimize translation edit rate (TER) on a tuning set. Preliminary results show that sentence-level weighting leads to significant MT accuracy improvements on the tuning set. However, the gains on the validation set are not any larger than just using corpus-level weighting, indicating the need of smoothing.

HierDec Improvements

Several enhancements to BBN's statistical machine translation system, HierDec, were implemented during this phase, listed below:

<u>Modeling enhancements:</u> we implemented context language models to exploit context information in rule application, which slightly improved Arabic-English and Chinese-English MT performance. We also introduced a new decoding feature based on the

length distribution of non-terminals in translation rules. The combination of these two features provided about 0.7 gain in BLEU in both languages.

Decoding speedups: we improved the efficiency of our rule extraction procedure, a major computing bottleneck in our MT experiments. The improved procedure reduces disk usage by more than 80%, total CPU usage by 25%, and wall clock time by 50%. We also worked on reducing the size of the rule file and improving the speed of the HierDec decoder, by restricting the admissible word spans in the generalization of non-terminals during rule extraction to individual word tokens and to certain types of syntactic constituents (e.g. adjectives and noun phrases) based on the target parse. Both techniques significantly reduced the size of the rule file and improved decoder speed with a small loss in performance for Arabic MT.

Lattice decoding capability: we implemented lattice decoding in BBN's HierDec system, which allows us to employ different tokenizations of the input to improve the coverage of translation rules. We observed slight improvement over string decoding under certain configurations on Arabic-English MT by combining word-level and morpheme-level tokenizations into a lattice. A small improvement was also obtained in Chinese-English translation, by decoding a lattice of multiple alternative character-to-word segmentations.

Manual translation rules: we incorporated manual translation rules in BBN's HierDec system. Some translation rules can be more accurately provided by humans than through machine learning. For example, quotation marks should always be translated as pairs. We implemented a mechanism that allows the decoder to use pre-specified translation rules to translate certain spans of input words. This provides a generic method to employ human knowledge in a statistical MT system. Experiments show this approach works well for translating paired quotes and parentheses.

Optimization for MT

Recently, a set of HTER annotation experiments were performed at BBN in order to determine the effect of tuning MT systems based on NIST-BLEU or IBM-BLEU. The two flavors of BLEU scoring use the same method to compute n-gram precision matches between the MT output and the reference translation, but differ slightly in the definition of the brevity penalty when the test data has multiple reference translations. It turns out that systems tuned using NIST-BLEU tend to generate shorter translation outputs compared to systems tuned using IBM-BLEU, which made us wonder whether tuning with IBM-BLEU could alleviate the known problem of content word deletions in our MT outputs, and perhaps give us a gain in HTER. Indeed, the HTER annotation experiments confirmed that there is a slight gain for tuning with IBM-BLEU.

System Combination for MT

We applied the improved system combination approach (with incremental system output alignment) on the Chinese retest (Phase 2.5) evaluation. Results on the development sets show that the AGILE P2.5 system combination output consistently outperforms the AGILE P2 system combination, due to improvements in both individual machine translation (MT) systems and system combination method. The gain is particularly significant on the newswire genre (2% absolute TER reduction and 3% absolute BLEU increase). Similar performance gains were observed on the Arabic P3 setup.

We explored various methods for aligning MT system outputs in order to construct confusion networks for system combination, including METEOR-based alignment (with stemming and WordNet synonym matching) and GIZA++ based alignment. So far, we have not seen any improvement over our incremental TER alignment method, but the resulting confusion networks appear to be more compact.

We investigated a cross-adaptation method for system combination, which incorporates source-side information. The implementation is based on language model biasing in the MT decoder. This method offers an alternative way for combining systems, although it has not yet been shown to be significantly better than the standard combination technique.

Sakhr-BBN Cascade

We have investigated various approaches towards a tighter integration of the BBN statistical and Sakhr rule-based translation systems, including constrained combination within noun-phrase boundaries, and BBN-rescoring of multiple Sakhr translation hypotheses. So far, the approach that works best is to first translate the Arabic text using the Sakhr system, and then automatically post-edit Sakhr's MT output using BBN's statistical MT system. We used the cascade system in the Arabic P3 evaluation as an MT component in the final system combination. This led to a significant improvement in the BLEU score (over 1 point on newswire and web), on top of the regular system combination, which already included the standalone Sakhr and BBN MT outputs.

4.2 Cambridge University

Context-Dependent Translation Models for Alignment of Parallel Text

We introduced alignment models for Machine Translation that take into account the context of a source word when determining its translation. Since the use of these contexts alone can cause data sparsity problems, we developed a decision tree algorithm for clustering the contexts based on optimization of the EM auxiliary function. The obtained context-dependent models led to an improvement in alignment quality and an increase in translation quality when the alignments are used to build a machine translation system, for both Arabic-to-English and Chinese-to-English tasks.

Efficient Strategies for Hierarchical Phrase-based Translation

A hierarchical phrase-based translation system was implemented following the k-best cube-pruning algorithm of Chiang. Two efficiency refinements intended to reduce both search errors and memory usage were developed (k-neighborhood exploration and smart memorization). A careful study and classification of hierarchical rule extraction into syntactic classes based on the number of non-terminals and the pattern led to filtering large number of hierarchical rules, achieving very efficient translation times with little degradation in performance.

Hierarchical Translation with Weighted Finite-State Transducers

A lattice-based decoder for hierarchical phrase-based translation was implemented with standard Weighted Finite-State Transducer (WFST) operations as an alternative to the well-known cube pruning procedure. We found that the use of WFSTs rather than k-best lists requires less pruning in translation search, resulting in fewer search errors, direct

generation of translation lattices in the target language, better parameter optimization, and improved translation performance when rescoring with long-span language models and MBR decoding. Reported translation experiments for Arabic-to-English and Chinese-to-English tasks showed this improved performance of the WFST-based hierarchical decoder in contrast to hierarchical translation under cube pruning.

Minimum Bayes Risk for Combination of Translation Output Obtained from Alternative Segmentations

Minimum Bayes Risk (MBR) offers a simple but very effective approach to system combination of translation output. Under this approach, N-best lists from multiple SMT systems are merged; the posterior distributions over the individual lists are interpolated to form a new distribution over the merged list. MBR hypothesis selection is then performed using sentence-level BLEU score in standard way. This was applied not only to combine our phrase-based system with the newly-developed hierarchical system, but also combining a single system trained on multiple segmentations/tokenizations of the data. In Arabic-to-English, we combined translation output generated by our hierarchical system when trained with MADA-tokenized and Sakhr-tokenized Arabic text, with significant improvement over the best of these two. In Chinese-to-English, we combined translation output generated by our hierarchical system when trained with Chinese text segmented by BBN within AGILE and by a segmentor and tagger developed at the University of Oxford. Significant improvements over the best individual system were also shown.

Participation in the GALE 2008 Machine Translation Evaluation

Cambridge submitted systems in all conditions for the Arabic and Chinese to English translation evaluation.

4.3 Information Sciences Institute (USC)

Over the last year, ISI has made many large and small improvements to the accuracy of our syntax-based, hierarchical-based, and phrase-based machine translation systems. Here we summarize major projects. All reported gains are for IBM BLEU on SysCombTune sets.

MIRA Training

A serious problem with standard MERT (Minimum Error Rate Training) is that it only scales to tens of features. Over the year, we tackled how to replace MERT with an efficient online learning algorithm. We implemented MIRA training, which not only overcomes the scaling problems of MERT, but also addresses the generalization problem through the use of large margin training. MIRA samples the decoder forest in order to find translations that receive high model scores, but are in fact bad translations (according to the references). It then seeks to punish these translations by adjusting the weights of a linear model that may contain tens of thousands of features. This lets us develop and exploit new feature sets to fix errors in our translations; our quarterly reports have detailed these features. In Chinese/English experiments, we obtained +1.1 BLEU on syntax-based MT and +1.5 on Hiero.

Heuristic Alignment

We have developed an expert program that aligns Arabic/English sentence pairs using a wealth of linguistic knowledge, including parse trees and a detailed analysis of how function words operate in translation. This program lets us fix alignment errors that systems like GIZA and LEAF fall prey to. It also lets us avoid slavish adherence to gold-standard alignment styles, which we have shown to be sometimes detrimental to machine translation performance. We have obtained +0.7 BLEU in our first trial with syntax-based MT.

Genre Tuning

By stratifying our language model resources and building genre-specific interpolated language models, we have improved our phrase-based translations by +0.2 in Chinese/English.

Forest-based Minimum Bayes Risk

We developed a fast method for calculating n-gram expected counts over an entire decoder output forest. These expected counts allow us to select a better translation from our n-best lists – we select the translation with the highest expected BLEU score. This technique improved Hiero translations by +1.0 (Arabic) and +0.4 (Chinese), and improved syntax-based translations by +0.1 (Arabic) and +0.2 (Chinese).

State Splitting

Our syntax-based system assembles English target trees by applying translation rules. Each rule produces a constituent in the tree, such as a NP (noun phrase) or a VP (verb phrase). We have found that, combination-wise, there are many types of English VPs -- for example, compare "went to the store" with "going to the store". We use an EM algorithm to split syntactic categories in our training data, replacing VPs by VP-0, VP-1, VP-2, and VP-3. When the decoder assembles English target trees, is now more likely to avoid ungrammatical structures. We obtain a +0.3 improvement for Chinese/English.

Syntactic Distortion

By punishing reordering over very large variables, we obtain +0.4 improvement for Hiero and syntax-based translation.

Arabic Spelling Correction and Morphology

We gained +0.1 by correcting typographical errors in Arabic source text. By using source-side lattices to encode morphological ambiguity, we previously obtained +0.5 in syntax-based translation; this year we obtained additional +0.3 by refining the method.

We also contributed to analysis that led to AGILE systems training on IBM BLEU, and we began new projects on more accurate parsing, faster decoding, synchronous tree-adjoining grammar, and error analysis.

4.4 Massachusetts Institute of Technology (MIT)

A TAG-based Discriminative Parser

We developed a new discriminative parser that was trained using the averaged perceptron algorithm. A key motivation for our approach is that it allows a great deal of flexibility in

the features which can be included in the model. The model combines a TAG-based grammar, which allows a rich set of features, with a coarse-to-fine dynamic programming approach based on a simpler dependency parser. A key step was to develop a TAG formalism that captures important dependencies in existing treebanks, but which can be parsed efficiently. The parser gives the best performance of any single-pass model (i.e., of any model that does not use reranking with a generative model) when applied to standard benchmarks for English parsing. The model will be useful in building full syntax-based translation systems.

A TAG-based Translation Model

A major focus in Year 3 has been on extending the TAG-based parsing algorithms to the translation task. We have implemented the following steps:

- We implemented the dynamic programming algorithms for parsing but over a lattice input. This will allow us to create a lattice of possible translations using a phrase-based approach. Each edge in the lattice corresponds to some substring of source-language words, and contains an English phrase together with some syntactic structure (TAG spines and some dependency attachments) associated with that phrase. The dynamic programming method will then search for the lowest cost parse-tree that spans the lattice. Some reordering will be allowed as TAG adjunctions are made, in a similar spirit to the ISI tree-transducer models of Marcu et al. (2006).

- We implemented code that extracts phrase-table entries that include English syntactic structure. This was a relatively simple step, using a simple augmentation of the MOSES phrase-table extraction method.

- We implemented code that takes an input sentence, and creates a lattice of possible phrase-translations, together with their associated syntactic structure.

- We implemented a syntactic language model that is used within the approach. The language model makes use of bi-gram lexical dependencies, and tri-gram lexical dependencies for "sibling" dependencies in the parse tree.

- We developed decoding algorithms that allow non-projective parsing operations. In the most general case, phrase entries can be permuted into any order before being combined to form a parse tree. In practice, we are investigating hard and soft constraints on these reordering operations.

In most recent work on this topic, we have developed efficient beam-search algorithms for the non-projective model; implemented a probabilistic discriminative model that links source-language features to target-language dependencies; and integrated a trigram language model into the approach. Preliminary experiments with the approach show small improvement (perhaps 0.4 to 0.8 BLEU points) over a regular phrase-based model, but there are many avenues for extending and improving the model, which we are currently exploring.

4.5 Sakhr Software

Sakhr Software performed three major tasks aimed at improving the overall performance of Arabic-to-English MT.

We began generating multiple translations out of the Sakhr MT system. This was done by preserving the output of certain paths in the translation process which are otherwise eliminated. These multiple translations were used in system combination experiments by BBN.

We performed human oracle experiments in the results of a cascade system of a Sakhr Arabic-to-English system followed by a BBN English-to-English (post editing) system.

We produced parse trees for a corpus of Arabic input sentences along with features that are used internally in the Sakhr MT system. These parses along with their associated features will be used by BBN as additional information inputs to the BBN SMT system that may help the resulting quality of Arabic-to-English translation.

4.6 University of Edinburgh

We further developed our statistical factored phrase-based machine translation system Moses and participated in the evaluation campaigns organized by the GALE project.

We carried out research along several axes.

Discriminative Training

We explored the use of discriminative training methods to optimize the millions of parameters of a statistical machine translation model both for the case of phrase-based and tree-based models. We showed competitive results with probabilistic models and explored the use of novel sampling methods.

Factored Translation Models

We developed an extension of factored translation models called factored template models, with small gains in mid-range reordering.

Efficiency

We demonstrated the use of suffix-arrays to efficiently store translation models, the use of randomized data structures to efficiently store language models, and the use of early discarding techniques to speed up the decoding process.

Tree-Based Models

We implemented a general framework of grammar-based translation models on top of the Moses that is capable of dealing with arbitrary non-terminal labels, and factored representation of non-terminals and terminals. This has reached a mature prototype stage and we will carry out extensive experimentation and optimization in the future.

4.7 University of Maryland

Translation Model Adaptation

We have extended our language adaptation procedure that takes advantage of comparable English documents to translation model adaptation. In this approach, new translation rules are automatically induced from English documents that are comparable to the source document, providing modest gains in TER and BLEU in Arabic-English translation, on top of language model adaptation. We are currently trying translation model adaptation on Chinese-English, and we are finding that it likely requires the use of a larger number of comparable documents than was used in Arabic-to-English translation. The use of a larger number of comparable passages requires refining the methods for selecting the set of possible bias rules being generated.

Paraphrasing for MT

We have applied automatic paraphrasing techniques to various areas, listed below:

<u>Regular parameter tuning:</u> we successfully ported the technique for automatically generating paraphrases to use the HierDec decoder instead of the phrase-based decoder. We then generated paraphrases for existing MT reference sets, which were used to optimize the feature weights of the MT system. Our results show no performance gain was achieved for the HierDec decoder, even though using the same technique yielded increased accuracy with the phrasal decoder.

<u>Tuning based on automatic post-editing:</u> we attempted to use the full-sentence paraphraser in a targeted fashion to produce a reference that is closer to each translation hypothesis in the system combination N-best list, without deviating from the meaning of the original reference. The motivation is to be able to use a better approximation to the HTER measure (which is used for downstream evaluation) as a tuning criterion. We are currently evaluating MT outputs optimized with this paraphrasing technique in terms of HTER.

<u>Source-side paraphrasing:</u> the goal of this project was to create new translation rules using source-side paraphrases. In decoding, we added a feature that represents the paraphrase probability of each rule (the non-paraphrase rules have probability 1.0). A source-side language model feature was also used, which compares the LM score of the source paraphrase to the LM score of the original source phrase, within the context of the test sentence. The use of the paraphrase rules did not show an increase over the baseline. Analysis showed that although the paraphrases generally seem reasonable taken out of context, they are unreliable to use directly in decoding.

Translation Rules with POS-tagged Non-Terminals

We modified the rule extraction process so that the non-terminals of hierarchical rules were set to be the part-of-speech tag of the source span that they covered. These parse rules were then added into the unmodified set of regular rules. During decoding, the non-terminals of the parse rules were compared against the parse of the test sentence, and only rules that matched the parse were used. When a given parse rule actually helps with part-of-speech disambiguation, it should have a naturally higher probability value, which encourages its use. In addition, we added an explicit decoding feature which is triggered when a parse rule is used. Unfortunately, we did not see a gain over the baseline using

the parse rules. Analysis showed that very few part-of-speech errors were committed in the baseline, so it seems this specific type of source-side analysis is unlikely to help translation.

Penalizing Deletion of Content Words

We had hypothesized that a major cause of MT "deletion errors" is the use of translation rules that have certain source words unaligned with any target words. We developed and tested a number of features to enable fair penalization of rules with unaligned source content words. Other features included the part-of-speech tag and the relative placement of the unaligned word within the rule. Unfortunately, we did not see a consistent gain in any of the conditions we tried. An in-depth analysis revealed that each feature caused roughly as many new errors as it fixed and that, overall, the decoding results were largely unchanged.

TERp-based Optimization

We developed an enhanced version of the TER scoring procedure that takes into account stemming, synonymy and paraphrasing when aligning the MT system output to the reference translation. We then examined using the new TERp metric to optimize the parameters of the HierDec translation system. A number of parameterizations of TERp were used to optimize the HierDec system for Chinese-to-English translation, and the results were compared against optimization using IBM BLEU as well as the original TER evaluation metric. A portion of the most promising results after optimization with TERp were annotated for HTER scores and compared against HTER scores after decoding using parameters optimized with IBM BLEU. Optimization using IBM BLEU was determined to have a noticeably improved, although not statistically significant, HTER score, prompting IBM BLEU to continue to be the standard metric used for parameter optimization for HierDec.

Lexical Translation Probabilities

The lexical smoothing score is a well known rule-level feature with the following intuition: "Does every word on the target side have a high probability match on the source side, and does every word on the source side have a high probability match on the target side?" We ran two experiments to explicitly measure the importance of the lexical smoothing feature compared to our standard phrase translation probabilities. To our surprise, removing the lexical smoothing feature hurt significantly more (reduction of 3.5 BLEU) than removing the phrase translation features (reduction of 1.5 BLEU). This indicates that we should pay more attention to the lexical smoothing feature than we do right now.

On-line Translation Rule Extraction

We developed a fast translation rule filtering method that allows the BBN HierDec system to run in on-line mode. The technique employs a simple, domain-specific inverted index algorithm that is able to scan a large, unfiltered translation rule file (150GB file containing 1.5 billion rules in the Chinese-English system) in approximately seven seconds in order to locate all applicable translation rules for a 60-word sentence, which is the longest sentence we would likely decode in a real-time system.

5 Accomplishments in OntoNotes

5.1 BBN Technologies

Distributed Two Major OntoNotes Data Releases

OntoNotes 2.0 (delivered to the LDC in November, 2007) added coverage of English and Chinese broadcast news (BN) data and initial coverage of Arabic newswire (NW) data, as well as extending the earlier English and Chinese coverage. OntoNotes 2.9 (delivered to the LDC in February, 2009) included English broadcast conversation (BC) data and additional Arabic NW. OntoNotes 3.0 will be released in the second quarter of 2009 (after the revised English parse trees for the English-Chinese Treebank are available from the LDC), and will include Chinese and parallel Chinese-English BC data.

Released Parallel Treebank Data

We distributed in March 2009 an English Treebank pre-release that included annotation for much of the Year 4 OntoNotes Web data as well as revised English trees for the existing OntoNotes data, as requested by the Banks Committee. The revised trees add NML constituents to capture the internal structure of the prenominal portions of noun phrases that used to be flat, and shift to a new standard that involves splitting almost all hyphenated tokens to make the constituent subtokens available for PropBank annotation. We also developed code to map our OntoNotes Treebank trees onto the original transcript files, determining the character offsets for each tree token.

Data Management and API Extensions

We developed code to map our Arabic annotation onto the revised ATB 3 parse trees released by the LDC. We extended the DB schema, API, and loading routines to support parallel Chinese-English data, including tree to tree mappings between the languages, and improved our routines for collecting and merging data for the different annotation layers automatically, reporting any clashes.

Continued Coreference Annotation

BBN continued its multilingual annotation of coreference, focusing on the Year 3 BC data in English and Chinese, and on additional NW in Arabic.

Coordinated Targeted Data Selection and Expedited Annotation Experiments

We worked with the other sites on targeted selection of Web data for Year 4, selecting documents rich in under-represented words and senses. BBN also provided trained word sense models to Colorado and ISI for use in exploring expedited annotation paths, for example, doing only single annotation of instances that an initial model classifies as the predominant sense with high confidence.

Experiments in using OntoNotes Data

BBN worked on various experiments in using OntoNotes data in GALE systems, including using propositional information to enrich the MT system's dependency parses, using word sense data to improve event extraction for Distillation, and using word sense information to improve word cluster models.

Planning for Year 4

Toward the end of the year, we revised our plans going forward based on input from the Program Manager to focus on completing our coverage of verb argument structures across all three languages.

5.2 Information Sciences Institute (USC)

Work at ISI focused on two aspects:

- Creation of senses for nouns and their annotation in the corpora,
- Pooling of noun senses to form concepts, and insertion into ISI's Omega ontology.

Creation of Senses for Nouns and their Annotation in the Corpora

Noun sense creation. This work, performed by a professional lexicographer in Boston (for English) and high-quality staff in Los Angeles (for Chinese and Arabic), has gone very well, except for a period of some 9 months for Chinese, when an interim sense creator, who has been fired, created senses that were inadequate and for which annotation agreement could not be reached.

Noun sense annotation. We delivered all the results to BBN in February 2009. Overall, annotation has required the services of about 25 annotators, all working part-time.

For **English**, we have: Total noun instances double annotated and adjudicated: 127891 (around 60% of total polysemous instances); Average agreement: 0.90. We have already handled more noun types than planned; but our noun instance coverage is not yet up to 80% because we have not yet completed a few very high-frequency nouns.

For **Chinese**, we have: Total noun instances double annotated and adjudicated: 32273 (around 18% of total polysemous instances); Average agreement: 0.95. Work lagged far behind schedule because of inadequate sense creation early last year; we had to discard several thousand annotated instances and re-do them after new sense creation. We identified and fixed the problem and were planning to start a new intensified effort for Chinese at the start of Year 4.

For **Arabic**, we have: Total noun instances double annotated and adjudicated: 6583. Average agreement: 0.89. Annotation is proceeding smoothly.

Active Learning. A postdoc visitor on sabbatical to ISI built an Active Learning system that we trained on the Year 1 corpus and deployed on the Year 2 corpus. We continued developing stopping criteria for the Active Learning procedure; a paper on this work was presented at the ICJNLP conference.

Annotation infrastructure. The SVN server that we installed last year continues to make data management easy, including versioning control for occasional rollbacks as needed.

Pooling of Noun Senses to form Concepts and Insertion into Omega Ontology

This work addresses the grouping of word senses into 'sense pools' that share the same meaning, and the taxonomization of the pools (from various languages) into a single ontology called Omega. We are continuing to build Omega 5, a new version of ISI's ontology, out of the word senses. This was delivered to BBN in February 2009.

Upper Model. In the past year we refined the Upper Model portion for Objects (noun-derived concepts and their generalizations) and worked with Colorado, who is developing the Upper Model portion for Events (verb-derived concepts and their generalizations). The former contains about 110 concepts, and the latter about 30.

Pool creation and insertion under the Upper Model. We employed four annotators to determine whether or not to group noun senses into 'sense pools', which, once ontologized into Omega, function as concepts. This process proceeds in tandem with sense definition, and involves both creating the pool and then inserting it into the appropriate point(s) under the Upper Model. Over 2100 pools derived from polysemous English nouns have been created and ontologized; these constitute the initial structure, which provides the framework for the rest of the work. We are currently working on the next 2800 terms, which are all derived from monosemous nouns (and for which no annotation is required). Sense pools have been created for them. This number covers the entire current corpus. We have another 7500 in reserve to be done later. A specialized interface and a procedure for identifying just the likely English-derived pools is being used.

Regarding the senses of Chinese and Arabic nouns, we have developed a variant of this procedure. The senses, once created and verified through annotation, are displayed in a specialized interface, together with the most likely candidate pools (derived from English) for them to be inserted into.

Outreach

We have developed a tutorial on annotation that uses OntoNotes as a working example, and presented it at several international venues. We have also published several papers on our work in OntoNotes, both in collaboration with other members of the team and on work performed at ISI alone.

5.3 University of Colorado (CU)

There are 12 separate GALE tasks in process at Colorado. In addition to adding Arabic sense tagging, and Chinese Sense Pooling and Ontology insertion, we are also doing Arabic Sense Pooling and Ontology Insertion for verbs.

An average inter-tagger agreement (ITA) of 89% is still being achieved for our delivered English, Chinese and Arabic sense tagged data. 2177 English verbs have been grouped. 1703 have received double annotation and adjudication on the WSJ, EBN, EBC and ECTB data (158,118 instances) with the 13 highest frequency verbs receiving only single annotation (17,900). We implemented the data selection plan for WebText documents. We tested a technique for using source language perplexity to automatically detect rare sense instances in a new corpus, which we are using for sentence selection to improve our performance on words without enough representative data. The supervised WSD system is currently achieving 85% on the 215 most difficult verbs with an average ITA of 85% and 91.13% accuracy on verbs with an average ITA of 92.81% (MFS baseline of 81.16%). The 160 Arabic verbs sensed in Year 2 have been double annotated and adjudicated, with additional data giving rise to extensive sense entry revision. Roughly 9000 tagged instances were delivered. 700 English verb pools have been added to Omega which is linked to over 30 nodes in the newly created Verb Upper Level

Ontology. These pools include 1127 English senses, 138 Chinese senses, and 44 new Arabic senses. *We have 87% to 89% coverage of verb senses in WSJ, EBN and EBC.*

The English BC and ECTB data (80.6K instances) have been PropBanked and delivered as well as the English Translation of the Chinese BC data (8K instances). 344 new framesets have been added, 254 for unseen verbs. *Our frame files cover over 99% of the verb instances in GigaWord and WebText.* We are waiting for the finalization of the revised ECTB Treebank before revising the PropBank. We upgraded the adjudication mode for the Jubilee PropBank annotation tool so that adjudicators can choose whether they want to skip instances where annotators agreed or view them. There is also a parameter which allows either two or three annotations per task. We have ported this system to Arabic.

The Chinese annotation project was restarted in mid March. 120K of the 150K of the Year 3 Newsgroup data has been pre-processed and treebanked. 30K is on hold because of character encoding issues that we are still struggling with. The 150K Chinese Broadcast Conversation has been PropBanked and is ready for delivery. The Chinese BC verb sense tagging of the 400 verbs sensed in Years 1 and 2 is done (19K), and 100 new verbs have been sensed and tagged in all the data, (20K instances).

5.4 University of Pennsylvania (Penn)

Penn this past year completed the parsing and full hand-correction of about 310K words of English text. Of this about 96K were texts translated into English from Arabic web materials (55K) and Chinese web materials (42K), enriching available GALE parallel text resources annotated for syntactic structure, and 72K were materials from web materials originally in English. Also parsed and hand-corrected with 140K words of P2.0 and P2.5 MT evaluation materials. This includes all unsequestered P2.5 materials, and all unsequestered P2.0 Chinese evaluation materials. In total, about 240K words of English materials were treebanked during Year 3.

Another major effort this year was bringing all materials previously treebanked by the Penn OntoNotes group and the English Treebanking team at LDC into conformance with a single annotation standard that differs, as the result of decisions by the GALE Banks Committee, somewhat from any existing materials. This new standard calls for a richer structure for NPs than Penn's older treebanked materials, and a new tokenization standard for words containing hyphens. Penn modified and hand-checked about 800K words of material previously treebanked for GALE or part of the OntoNotes database. An additional 500K of WSJ materials were modified, yielding a total of 1.3 million words of material from Penn consistent with the new standard.

During Year 3, Penn also began experiments on annotator accuracy for English Treebanking that will continue in Year 4 and now in collaboration with LDC. In an initial experiment, we re-annotated about 6K words of BN materials that had previously treebanked about 1.5 years earlier. Viewing one year as gold standard and the other as test set, labeled bracket recall was 98.4, labeled bracket precision was 98.5, and tagging consistency was 98.7.

In other work that will continue into Year 4, Penn has begun to investigate discriminative methods for idafa placement in Arabic, with two papers published, we have developed a

new discriminative model for placement of null elements in Arabic and English text that explicitly models the syntactic sub-categorization structure of the verb and chooses null element placement given the options. Initial results on gold standard material range from 99.5 F on WHNP 0 null elements to 73.0 F on adverbial Wh-elements. The crucial category of Nominal WH-traces is 87.5 F, about 10 F points lower than English.

6 Accomplishments in Distillation

6.1 BBN Technologies

Single-document Question and Answering

We developed a new system to answer specific questions from particular documents.

Work for this task included:

- Modifying the existing Distillation system for the particular focus of a single-document task (especially the requirement of exact answers rather than snippets)
- Enabling full distillation in the source language and developing methods to intelligently combine information derived from both source language and MT
- Improvements to proposition trees, including their extension to Chinese (using propositions) and Arabic (using modified parses), and variations and relaxations on tree-matching which still capture essential structural relations
- Developing an annotation tool for the task
- Template-specific development to improve performance across the board

5W Identification

We developed a new system to identify the WHO, WHAT, WHEN, WHERE, and WHY (the 5W) of a sentence. We were also involved with the task definition for this system.

Work included:

- Statistically modeling root predicate selection from sentences
- Using alignments to the source language to filter less-likely MT predicates
- Using n-best translations to allow system to mitigate MT errors
- Using semantic role labeling and propositions to identify W arguments in the source language

We participated in the Phase 3 evaluation and passed all go/no-go targets.

Distillation-biased MT

We used Distillation input and/or output to bias the MT system and to improve the usability of its output for downstream processing.

For the Phase 3 Distillation task, we used user queries to bias the translation of a single document, both improving overall performance and creating translations more likely to be understood as the answer to a specific question.

For the Phase 2 Distillation task, we used Distillation system output over English documents to bias the MT system to produce output more similar to those (hopefully relevant) English snippets, improving overall Distillation recall and no cost to precision

Improvements to Corpus Based Query-response System

We modified the existing system to run periodically on a document feed (rather than from a fixed corpus). We also decoupled the system representation of response snippets from propositional node spans, moving to a less-noisy representation of complete and partial text span 'nuggets' to improve our redundancy detection output.

Other Work

We applied unsupervised techniques to construct domain-restricted word similarity probabilities, specifically targeted for use in template-based question-answering.

Framework for the creation of a redundancy dataset was developed and was used for experiments in redundancy removal.

We also participated in various pilot efforts to help define the Distillation Phase 3 task.

6.2 Information Sciences Institute (USC)

We have created and made available for public distribution the automated summarization evaluation tool BEwT-E; see http://www3.isi.edu/services/services-licensable_software.htm. This tool was used in the international TAC evaluation conference organized by NIST in 2008. We have also created a preposition disambiguation system that outperforms all existing systems in a standardized test.

Nuggets for Distillation Tasks

The automated nugget (eNug) creation system breaks sentences down into very small Basic Elements (BEs) of various kinds that can be recomposed into whatever nuggets are defined. Being of small size, these BEs can be more effectively canonicalized using various transforms, which allows them to be compared against other formulations of effectively the same content.

In order to perform automated evaluations of summarization or distillation output, the system accepts on the one hand a system text (summary or answer) and on the other one or more gold standard texts for the same test, typically produced by humans. The system breaks both texts down into a set of BEs, applies canonicalization transforms, and then compares the standardized lists. A higher overlap of BEs between the test and the gold standard BE lists means a higher agreement of elemental units of content, and thus a higher overall score.

We are pleased to note the following:

- BE correlates better with Pyramid than with Responsiveness scoring (i.e., it is a more precise, careful, measure)
- BE generally outperforms ROUGE (i.e., its transformations—the essential differentiator between the two systems—are working)

- Compared with ROUGE, BE's correlation improves on summaries created by humans (columns labeled 'Hu') compared to those created by systems (columns labeled 'Auto').

We entered the BEwT-E system in the Metrics-MATR MT evaluation system workshop organized by NIST in August 2008. While the system did not win (it came in around fourth, of over ten systems), we were pleased, since it was not tuned for MT evaluation and was a very young system compared to the others. Findings show reasonably high correlation between the human assessors and the BE engine (more details can be found in the paper we presented at the workshop). We also applied BEwT-E to NIST's TAC Update Summarization task.

Preposition Disambiguation

Two students at ISI built a preposition disambiguation classifier. We measured the accuracy of the classifiers over the test set provided by SemEval-2007 and provided these results in the table below. It is notable that our system produced good results with all classifiers. Using our extended parse-based features, all our classifiers except for kNN outperformed all three SEMEVAL-2007 contestants and the baseline.

7 Accomplishments in Integration and Operational Engines

7.1 BBN Technologies

The primary goals for the AGILE Operational Engines and Integration for the third year of the GALE program were the following: complete COTS integration of the Audio Monitoring Component (AMC) product for Arabic and Chinese using the GALE algorithms and models integrated into the engines in the previous year; integrate and validate the syntax-based Machine Translation engine from Language Weaver together with the AMC on a single-server for operational environments; and continue the integration of GALE Year 3 models and algorithms into the AGILE operational engines to enable rapid transition of GALE technology into the field. BBN was successful at three objectives and made significant progress in transitioning media monitoring products integrated with GALE technology to operational environments to contribute to the mission of the US Government of open source intelligence acquisition and analysis.

In the second year of GALE, we had completed the integration of research STT and MT algorithms and models into the operational engines for Chinese and Arabic. Last year we completed the configuration and update of the natural language processing components, the sentence boundary detection engine, and the named-entity recognition engine, to enable the COTS integration and release of the AMC product. We completed the operational integration of the new Perceptron-based discriminative version of the named entity recognizer (NER) component into the AGILE operational engine. We also benchmarked the name recognition performance gains that we achieved using this new component and demonstrated 15-25% relative gains in NER performance on Chinese and Arabic speech/text. We have also updated the sentence boundary detection (SBD) component for Chinese to match the new GALE-enabled STT. BBN made a significant concurrent investment in the AMC product to enable many engineering improvements such as support for single-server operation without dependence on enterprise-level

database, unified HTTP-based I/O interface, support for natural throughput of 0.5xRT, and support for 3 additional languages: Spanish, Farsi, and U.S. English. The BBN AMC 3.1 was finally released in August 2008.

BBN also made significant improvements to the STT models and algorithms in the AMC 3.1 during the third year of GALE. We successfully integrated online speaker adaptation into the STT engine. The adaptation procedure is compatible with the latest algorithms in the STT engine and can be configured to meet the operational requirements of the end-to-end system. We demonstrated that online speaker adaptation leads to 3-4% relative reduction in word error rate (WER) on Arabic and Chinese speech test sets. We also identified research STT models and algorithms from the GALE Phase 3 and Phase 3.5 evaluations for Arabic and Chinese, respectively, to integrate into the operational engine. We selected the best single Arabic grapheme-based research STT model and demonstrated early on that, even though it does not currently meet the AMC operational requirements, it leads to a relative WER performance improvement of 20% as compared to the COTS AMC 3.1 engine. We are continuing to tune the operational engine that includes the P3 Arabic model to meet operational requirements of speed, memory, and latency. For Chinese, we have identified the best single research STT model and have started the work of integrating it with AMC.

During Year 3 of GALE, BBN collaborated closely with Language Weaver for the development and testing of the functional and operational features of the syntax-based machine translation engine for speech input. LW finished the productization and release of the syntax-based MT v5.0 in October 2008. This engine included support for XML-based interface and cross-MT name linkage to allow integration with the BBN AMC. We also successfully demonstrated the integration of this new syntax-based MT engine for Chinese-to-English translation into the AGILE Operational Engine for single-server operation. We have benchmarked the translation performance and documented the runtime characteristics of the updated operational engine that includes the syntax-based MT engine. The integrated engine provides an improvement in translation performance of up to 5 BLEU points on the broadcast news speech data sets. The integrated end-to-end system also satisfies the specified operational requirements of the AMC Chinese product. We also started work on the integration of the LW Arabic-English syntax-based MT with the Arabic AMC. The integrated system meets functional requirements including XML-based interface and name linkage. BBN and LW collaborated during the year to identify additional machine translation quality settings that would allow the integrated operational engine to meet requirements of throughput, memory, and latency without significant loss of performance. The integrated operational engine that includes LW syntax-based MT v5.0 for Arabic does not meet performance or operational requirements but Language Weaver is continuing to improve speed and performance of the syntax-based MT engine to support this effort.

One of our goals for the Year 3 effort was to develop a capability for domain adaptation in the operational engine. Currently operational engines are trained on fixed corpora and once deployed they provide no mechanism for model update or customization. This lack of domain adaptation facility limits adoption of these systems. This year we focused on developing a capability in the AGILE operational engines to allow users to customize the lexicon of AMC to meet operational needs. We defined the initial approach during the

year as the development of a capability to enhance the Arabic AMC such that users can add new phrases to the STT lexicon without destabilizing the AMC in terms of operational characteristics. Our focus was on general phrases and not just names mainly due to the inherent difficulty in the conceptual definition of names as demonstrated by our previous work on name translation in the previous years. We defined the functional capabilities of the lexicon customization feature in the AMC and finished the annotation of names, according to MUC guidelines, in four GALE speech test sets: Dev06, Dev07, Eval06, and Eval07. We also proposed a two-part metric to measure progress: improved recall for user-added phrase tokens with each iteration; no degradation of overall STT WER performance compared to baseline COTS AMC. BBN is now ready to start experiments to investigate algorithms and techniques to enable the lexicon customization capability in the AGILE operational engine.

Contributions to the GALE Community

BBN set up four external servers in the second year of GALE that ran AGILE Year 1 operational engines for Arabic and Chinese to support LDC data collection and annotation work. A methodical analysis of the processing steps at BBN and LDC, in the third year, revealed that there were significant discrepancies between the two and this was leading to mistakes in data selection and annotation especially for GALE evaluations. BBN designed, implemented and delivered to LDC a simplified, robust framework to process audio data through the BBN-hosted AMC system to ensure that the two sites use identical process for this purpose.

Technology Transition and Transfer

During Year 3 of GALE, BBN shipped and deployed a two-channel Arabic BMS 3.0 system to CJSOTF-AP in Balad, Iraq. BMS 3.0 includes AMC 3.1 which was developed and released during the same year and which contains GALE research STT/MT improvements for Arabic and Chinese. This upgrade fields the one-year archive capability requested by the unit and developed under funding by DARPA. BBN paid for productization of the capability, license fees for the upgrade, shipment, and installation, and overseas training for the on-site personnel. This unit represents the first fielded instance of BMS 3.0.

BBN delivered and installed a two-harvester Web Monitoring System (version 1.4) supporting English and Arabic to USSOCOM headquarters in Tampa, Florida, under the TSWG GMT (Global Multimedia Tracking) effort. BBN also deployed a single-channel BMS 3.0 system as the first phase to upgrade USSOCOM's five-channel BMS 2.0 system. This system will be used for Information Assurance accreditation purposes in order to qualify BMS 3.0 for deployment on USSOCOM's networks.

BBN delivered 26 licenses for the BBN Audio Monitoring Component (version 3.1) to SAIC, Inc., for development and integration on behalf of an undisclosed Intelligence Community customer. This delivery consisted of AMC licenses across all 4 languages.

BBN delivered two turnkey systems of the BBN Audio Monitoring Component (version 3.1) to the Carnegie Mellon University Software Engineering Institute (SEI). The two channels are configured for English audio data.

7.2 Language Weaver (LW)

LW v5.0 Arabic-English Phrase-Based System Release

We released LW v5.0 Arabic-English Phrase-Based release with significant translation quality improvements over the previous release.

Test Set	LW v4.1 BLEU	LW v5.0 BLEU
2006 broadcast news tuning set	15.97	19.56
2006 broadcast conversation tuning set	11.56	14.35
2006 broadcast news test set	14.99	18.38
2006 broadcast conversation test set	12.86	16.36

LW Chinese-English Syntax-Based System Release with X-SPT support

We ported the X-SPT interface to the Syntax-Based system framework and we released the Operational Chinese-English Syntax-Based system leading to significant translation quality improvements:

Test set	LW 4.3	LW 5.0 quality 3	LW 5.0 quality 4
bnmd06	11.04	12.85	15.55
bnmt06	12.63	14.12	17.31
bcmd05	9.42	8.80	10.19
bcmdr06	6.04	5.06	6.26

LW Arabic-English Syntax-Based System Productization with X-SPT support

We productized the Arabic-English Syntax-Based system with X-SPT support. We released a Beta version that shows improvement over the Phrase-Based system on the newswire genre. The speed of the Syntax-Based system is, however, still an issue.

System	BLEU	Speed
Arabic-English Phrase-Based Q3	39.95	2215wpm
Arabic-English Syntax-Based Q3	41.01	232wpm

8 Appendix

This appendix contains the accomplishments for three additional efforts: Serif Maturation, Broadcast Monitoring System One-Year Archive, and Robust Automatic Transcription of Speech (RATS).

8.1 Serif Maturation

The primary goals of the Phase 3 Serif effort were to improve BBN's Serif information extraction system in terms of speed, accuracy, modularity and robustness, resulting in an updated release to the sponsor in September 2008. This effort was conducted in cooperation with the STTI Serif Maturation effort, which addressed identical goals.

BBN's syntactic parser had been identified as a significant bottleneck to the overall system speed and previous optimization attempts had not yielded enough throughput improvement. Under this effort, we contributed to a significant increase in overall Serif throughput by developing heuristics for determining which sentences have little to no extraction content, allowing the parser to bypass those sentences. We also experimented with a simplified part of speech model that led to improved performance and speed in our fastest Serif configuration, which relies on an NP chunker, rather than a full syntactic parser.

To address the sponsor's requirements for system modularity, we implemented and tested a new cross-platform general pipeline framework, known as "Modular Serif." This framework provides a set of public component interfaces and a generalized data structure for storing extracted information. For increased robustness, we integrated the new Modular Serif system into BBN's existing Server Byblos pipeline framework, which handles server invocation, error logging and automatic recovery from failure.

In support of achieving this project's goals, we created an improved development infrastructure that allows us to more easily conduct regression, throughput, robustness and longevity testing. In addition, we developed an automated release proccss.

We delivered the English Serif v4.0 release to the sponsor on 24 September. All efforts after that point were focused on additional testing and documentation, as well as responding to sponsor requests for support.

8.2 Broadcast Monitoring System One-Year Archive

In 2006, DARPA purchased a two-channel Broadcast Monitoring System (BMS) for deployment to the 5^{th} Special Forces Group in Balad, Iraq. The system was installed in December 2006. In November 2007, DARPA, TSWG, and BBN participated in a conference call with the users at Balad; during that call they expressed a requirement for a longer archive – their current BMS version v2 allowed storage of 90 days of video and associated metadata.

In 2008, DARPA provided funding to extend the BMS product with the capability to store one year of video, transcriptions and translations. BBN invested engineering funds concurrently to develop and release the new version of the BMS product (version 3) with the new enhancements.

BBN completed the development of BMS v3 in September 2008. The new version of the product includes the following capabilities:

- Store and search across one year of media and metadata archive on each channel.
- Integrated COTS Audio Monitoring Component v3.1 with GALE-Y2 Speech-to-text (STT) and Machine Translation (MT) research improvements.
- Support for MPEG-4 video encoding that leads to smaller media file sizes for extraction and archiving.
- Consolidated media archive on the video server component that now includes the metadata database in addition to the video files.
- Support for non-destructive BMS upgrades by isolating operating system drive from data drives on the video server.
- Support for 5 languages: Arabic, Chinese, Farsi, Spanish, and English
- Security support (DISA Gold and Retina eEye compliance)

BBN has also developed improved release testing procedures and tools to thoroughly test the BMS v3 product prior to deployment to the 5^{th} Special Forces Group.

BBN made a one-time offer to customers with current maintenance agreements and hardware to upgrade their systems to version 3 at no cost, in order to facilitate the transition of GALE enhancements to operational use. For customers without current maintenance, BBN proposed to upgrade systems for the cost of one year of maintenance and the purchase of required hardware. The Balad customer's maintenance expired in December 2007, and, despite working with them and a number of potential sponsors, BBN was unable to obtain funding for maintenance.

To enable the GALE transition, BBN management decided to assume the costs of shipping, installation, and training, with the hardware used in the development effort delivered as Government Furnished Equipment. Language Weaver also agreed to waive its annual maintenance fee and to provide a no-cost update to the machine translation software. BBN shipped three full channels of BMS v3 hardware (two operational and one spare) on February 13, 2009.

BBN's contractor for overseas installations spent the week of February 23rd, 2009 on site performing the upgrade. BBN provided training documentation and phone support throughout the week. We received confirmation from our overseas contractor in the first week of March, 2009 that the 2-channel BMS v3 system was operational in Balad, Iraq.

8.3 Robust Automatic Transcription of Speech (RATS)

This is the final report for the RATS seedling effort. The objective of the RATS effort was to assess the state of the art in performing various types of processing on real-world speech data of interest to the Government and to suggest approaches for future work aimed at improving performance. The types of processing include: speech activity detection (SAD), language identification (LID), automatic speech recognition (ASR), and key word search (KWS).

The report starts by describing the data that was provided to us by the Government under this seedling. That is followed by the results of experiments performed by BBN and MIT Lincoln Laboratory in each of the four activities: SAD, LID, ASR, and KWS. We then describe the English amateur radio speech data that we found online and the result of a preliminary baseline speech recognition experiment on that data. Finally, we summarize possible future methods of research for improving speech processing performance on these types of noisy speech data.

Government-Furnished Data

We were provided with an audio data set totaling about 6.5 hours of classified radio transmissions in four languages: Iraqi Arabic, Farsi, Dari, and Pashto. Table 8-1 shows the amount of speech that we received in each of the four languages. The audio was originally sampled at either 8kHz mono or 44.1kHz stereo but was then down-sampled and converted to 8kHz mono for further processing at BBN and MIT Lincoln Lab. The data was hand-segmented at BBN into speech and non-speech segments, to support further experiments. Details of the speech and non-speech segmentation are shown in Table 8-1. Overall, about 75% of the duration of the audio is speech while the remaining data is non-speech, with an average length of a non-speech segment of about two seconds. However, the speech proportion (in percentage) for Arabic Iraqi and Farsi is about 65% while it is almost 90% for Dari and Pashto.

Language	Total Duration (hours)	Speech Proportion (%)	Number of Speech Segments	Average Non-speech Segment Duration (seconds)
Arabic	2.14	64.1%	945	2.9
Farsi	1.43	67.4%	741	2.1
Dari	1.50	88.1%	706	0.9
Pashto	1.38	88.4%	540	1.1
OVERALL	**6.45**	**75.6%**	**2932**	**1.9**

Table 8-1: RATS corpus statistics.

The segmented data was then transcribed by companies in the Arlington, VA, area that provide transcription services using cleared personnel. BBN provided the software transcription tools, transcription guidelines in the four languages, and personnel to train the linguists in performing the transcriptions. Initially, one linguist from SOS International (SOSi) performed a first-pass transcription of the Iraqi Arabic data at BBN's offices in Arlington, VA, because SOSi lacked the facilities for doing the classified transcriptions in house. Afterwards, we utilized linguists from ASET International because the company had ready access to cleared linguists who were able to do their work at the company location and in the four languages, although some work was also done at BBN in Cambridge using linguists supplied by ASET.

For each language, we split the data into two equal subsets, one for training/development and the other for testing.

Speech Activity Detection (SAD)

We performed SAD experiments on the test set at BBN and Lincoln. We followed NIST's RT05S Meeting Speech Activity Detection evaluation guidelines in terms of creating SAD reference, using NIST's scoring tool, and reporting SAD error rate. Specifically, the SAD error rate is defined as the ratio of the total error duration to the total reference speech duration. Total error duration is the sum of missed speech duration and inserted speech duration (or false alarm).

The initial SAD experiment using BBN's Audio Monitoring Component (AMC) speech/non-speech classifier, which was trained on broadcast news data, produced a result of 15.7% SAD error. BBN then trained a simple classifier using Gaussian mixture models (GMM) on the training set of the RATS data; the result was a total SAD error of 4.9%.

SAD experiments were also performed at Lincoln Lab; Table 8-2 shows the results. The first two rows show the results using two off-the-shelf SAD models: the first set was trained on unclassified telephone speech and the second were trained on classified telephone speech. The total SAD error rates were 22.6% and 17.0%, respectively. The last row shows the results of training SAD models on the RATS training data using a more sophisticated classifier based Gaussian mixture models; the total SAD error rate was 4.6%.

MODELS	Miss (%)	False Alarm (%)	Total (%)
Unclassified Telephone Models	22.4	0.2	22.6
Classified Telephone Models	16.8	0.2	17.0
RATS Models	3.0	1.6	4.6

Table 8-2: Speech activity detection results obtained at Lincoln Lab with Miss and False Alarm obtained relative to the total duration of speech present.

The above results were obtained by dividing the Miss duration and the False Alarm (FA) duration by the total reference speech duration, for a particular setting of the Lincoln system. However, because the amount of nonspeech in this data was so small, the FA rates were unrealistically low. So, FA rate was then redefined as the ratio of the total FA duration to the total *non-speech* duration, which is appropriate for a detection task. ROC curves were then plotted and operating points close to equal error rates (between Miss and FA) were chosen to give the results in Table 8-3.

MODELS	Missed Speech Time (s)	Total Speech Time (s)	Miss (%)	False Alarm Speech Time (s)	Total Non-Speech Time (s)	False Alarm (%)	Total Error (%)
Unclassified Telephone Models	508.13	65.11.00	7.8	148.25	1891.17	7.8	15.6
Classified Telephone Models	651.04	6511.00	10.0	136.70	1891.17	7.2	17.2
RATS Models	326.57	6511.00	5.0	112.49	1891.17	5.9	10.9

Table 8-3: Speech activity detection results obtained at Lincoln Lab, where Miss is relative to amount of speech present and False Alarm is relative to the amount of non-speech present.

Language Identification (LID)

The language ID task was done completely at Lincoln Lab using a system based on Support Vector Machines (SVM) with Generalized Linear Discriminant Sequence (GLDS) kernel. Due to the small size of the RATS corpus, the LID experiment was

carried out using a five-fold cross-validation procedure. The corpus was partitioned into five folds (i.e., each language was divided into five parts). Each fold consisted of 80% of the data for training and the other 20% for testing. LID scores of each fold were pooled together to measure LID performance.

The full set of results was provided directly by Lincoln Lab to DARPA. Here, we report on LID performance as measured by the Equal Error Rate (EER) – the performance when the miss probability equals the false alarm probability. The EER was 14.9% for Arabic, 12.5% for Farsi, 3.6% for Dari, and 18.2% for Pashto, for an average LID EER of 10.9% for the four languages. The differences in performance may have been more a function of the properties and variability of the channels over which the data was captured, rather than a function of the specific language.

Automatic Speech Recognition (ASR)

The ASR work was performed completely at BBN. Most of the recognition experiments were performed on the Iraqi Arabic data, with some on the Farsi data. Results of the Arabic experiments are shown in Table 8-4.

The initial baseline Iraqi Arabic ASR system was trained on only TRANSTAC data – a narrow domain of conversations between soldiers or medics and civilians in a limited range of settings such as military checkpoints, door-to-door searches, or first-aid situations. The acoustic models were trained on 500 hours of Iraqi audio after being down-sampled to match RATS data in sampling rate. The language models were estimated using the 3M-word TRANSTAC text corpus. The recognition word error rate (WER) on the RATS Iraqi Arabic subset, using this initial (out-of-domain) ASR system, was 87.3%.

There was some improvement in WER after domain adaptation of the initial system to the RATS training data. Specifically, we added new words found in RATS data to the recognition dictionary (to lower out-of-vocabulary rate), interpolated the TRANSTAC language model (LM) with the small LM estimated from the transcript of the one-hour RATS training subset, and adapted the acoustic models using RATS audio data. The WER after adaptation was 79.2%.

In order to see whether training on a small amount of in-domain data would be any better than training on larger amounts of out-of-domain data, we trained our ASR system on the one hour of RATS training data, with a language model that used the RATS training data interpolated with the 3M words of TRANSTAC data. The result was a WER of 90.0%, clearly showing that training on too small an amount of in-domain data cannot match the performance of models trained on larger amounts of out-of-domain data.

Configuration	Acoustic Model Training	Language Model Training	WER
Baseline	500hr TRANSTAC	3M-word TRANSTAC	86.1%
Adapted	500hr TRANSTAC adapted to 1hr RATS	3M-word TRANSTAC interpolated with RATS training data	78.1%
RATS hybrid	1hr RATS	3M-word TRANSTAC interpolated with RATS training data	90.0%

Table 8-4: Iraqi Arabic ASR experimental results for three different configurations.

For the Farsi data, we built an out-of-domain ASR system using 80 hours of TRANSTAC Farsi corpus and 45M-word Farsi web text. The WER on the RATS Farsi test set was 90.2%. Adaptation to the Farsi data would have yielded some improvement over this result, but that experiment was not performed.

For Dari and Pashto, the only training data available to us was the small amount of RATS training data. Training on such a small amount of training data, for both the acoustic and language models, would have yielded even higher error rates than for Farsi. So, those experiments were not performed.

Key-Word Spotting (KWS)

We also performed keyword spotting (KWS) experiments with the Iraqi Arabic data using query terms that we chose after examining the transcripts of the Iraqi Arabic data. We chose 42 terms, most of which were spoken numbers (used as caller IDs during the radio conversations). The KWS system was based on the BBN system that had the best results in NIST's 2006 Spoken Term Detection (STD) Evaluation. This system uses word lattices with confidences from recognition to generate potential detection records. To measure performance, we used NIST-provided scoring tools to generate the probability of miss and probability of false alarm.

We carried out keyword spotting experiments with two ASR system configurations: the 'baseline' system with 86.1% WER and the 'adapted' system with 78.1% WER, as shown in Table 8-4. In Figure 8-1, we plot the probability of miss, P(Miss), against the probability of false alarm, P(FA), for the two ASR configurations. From Figure 8-1, we see that, at low false alarm rates, the baseline system has slightly lower P(Miss) than the adapted system, while at false alarm rates higher than 0.03, the adapted system has significantly lower P(Miss).

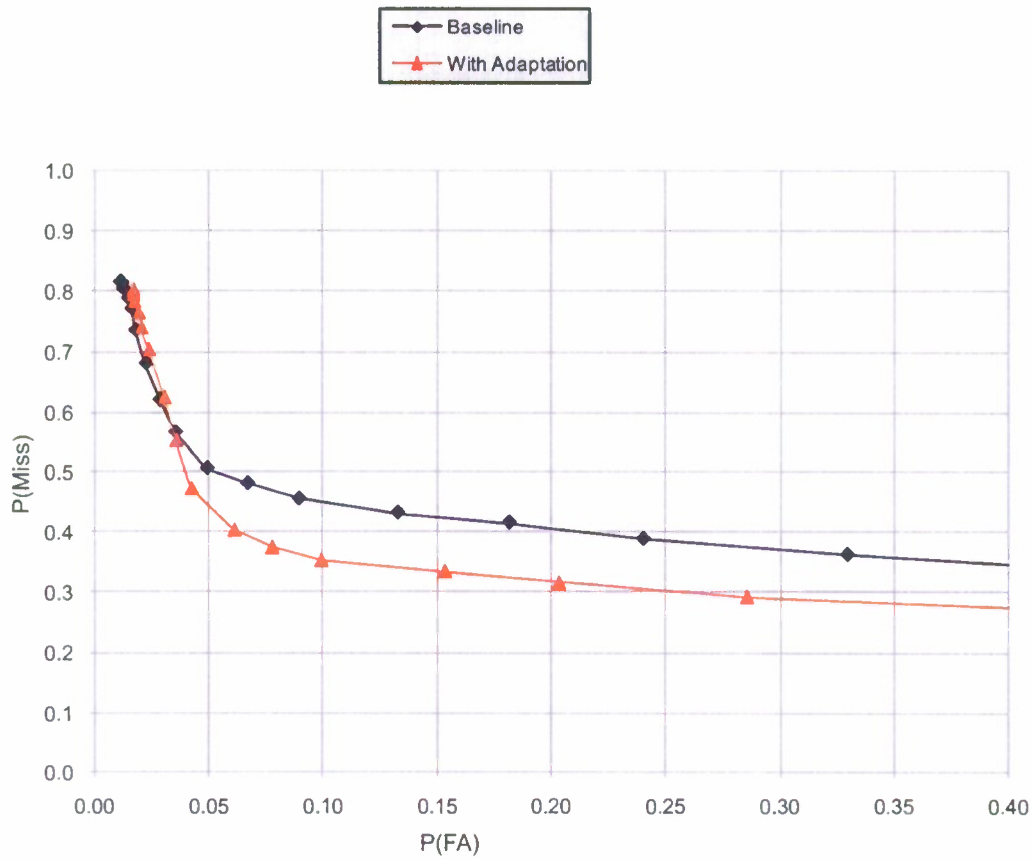

Figure 8-1: Comparing keyword spotting results on RATS test set using ASR outputs produced by the 'baseline' and 'adapted' configurations shown in Table 3.

English Amateur Radio Data

For any future RATS program, it would be good to perform research on open-source, real-world data that is as similar as possible to the contemplated classified data. We, therefore, made an effort to find radio speech data that is publicly available. As a result of that search, we found a source of amateur radio (aka ham radio) data in English on the web. In total, we found 114 hours of ham radio speech. Of this data, we chose and transcribed a test set that comprised 30 minutes of speech from eight audio files selected randomly, for a total of two hours.

In order to establish a baseline for performing ASR on this data, we used BBN's existing English broadcast news (BN) system as is. The system uses acoustic models trained on 1600 hours of BN data and language models estimated on a text pool of about 10 billion words. The typical WER of this system on BN test sets is 10%. In this study, we up-sampled the ham radio data to match the characteristics of the wider-band BN data. The overall WER was 60%. We would expect this error rate to drop if training or adaptation on the ham radio domain is performed.

Approaches to Dealing with Noise

During Dr. Joe Olive's visit to BBN, we made a presentation in which we laid out the framework for developing technology to deal effectively with noise to improve ASR performance on RATS data. In addition to simply training on more transcribed data, we outlined the following possible general directions for research:

1. Remove the noise: In this approach, an attempt is made to recover the original speech signal through spectral subtraction and/or de-reverberation. In all cases, it is best to retrain the system on examples of noisy speech that has been processed.

2. Discount the noise: This approach benefits from the observation that the effect of noise is greater at certain times and certain frequencies. It depends on being able to estimate the level and spectrum of the noise on a continuing basis. Then, at each frame, and each frequency, the contribution of that frequency to the overall recognition can be weighted appropriately, giving a lower weight where the signal is likely to be more affected by the noise. This approach requires the use of a frequency-domain probability model.

3. Unsupervised training: Recent research at BBN has shown that, with the availability of a small amount of transcribed data and a large amount of untranscribed data, it is possible to use unsupervised training methods to improve recognition accuracy significantly. It would be interesting to try out this approach to dealing with large amounts of noisy data similar to that contemplated for the RATS project.

4. Joint model of speech and noise: Given a model of clean speech and a model of current noise, this approach attempts to synthesize a model of speech-plus-noise to perform the recognition. The synthesizing procedure will depend on the type of noise.

5. Noise-robust feature transformation: The idea in this approach would be to perform a transformation on the input features or on the speech signal itself that results in features that are less sensitive to various forms of degradation. The transformation would be estimated from a wide variety of conditions and languages.

8.4 Serif Research

Overview

The primary accomplishment under the Serif Research funding was the development of an unsupervised system that learns to recognize specific relations (such as "X employs Y" or "X is a parent of Y") and concepts (such as "X is a person" or "X is an invention") by finding natural language patterns used to express them.

The system (called "LearnIt") begins with a small set of seed examples, or a small set of hand-constructed language patterns. It then alternately uses the set of known seed examples to search for new patterns; and uses the set of known patterns to search for new seeds. Each of these new patterns and seeds is evaluated, based on its consistency with the existing seeds and patterns; and only the new patterns and seeds that are most likely to be correct are retained.

Since the LearnIt system only requires a few example inputs to get started, it can be used to quickly learn patterns for finding instances of new relations (or existing relations in new domains) without requiring any annotated training data.

This final report will describe the overall design of the LearnIt system (0), discuss improvements made to the system in the last quarter of the year (0), and summarize the results of two LearnIt evaluations, one previously reported (0) and one new in the last quarter of the year (0). It also briefly summarizes the results of a previously reported experiment calculating an upper bound for performance in a low-resource language compared to performance over low-resource MT (0).

LearnIt System Design

The overall architecture of the LearnIt system is summarized in Figure 8-2. We begin with a set of example seeds or patterns from the user, and then iteratively propose patterns, vet patterns, propose seeds, and vet seeds. As we iterate through this cycle we build up a set of seeds (each of which has an associated score and confidence) and patterns (each of which has an approximated precision and recall score). Patterns are expressed using the Brandy Pattern Language, a pre-existing pattern language that was developed for answer selection in the GALE Distillation project. This pattern language operates over surface text, propositions, ACE events/relations, or any combination thereof, and includes support for regular expressions.

A detailed view of the LearnIt system's workflow is shown in Figure 8-3. Each column represents a single "stage" of the system, and corresponds with one shaded box in Figure 8-2. As is clear from the figure, each stage has a similar overall structure; they differ mainly in what they search for; and in what they do with the seed matches or pattern matches at the end of the stage.

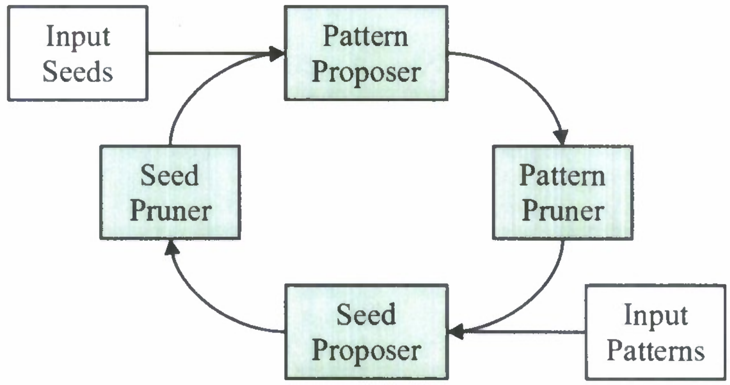

Figure 8-2: LearnIt System Overview

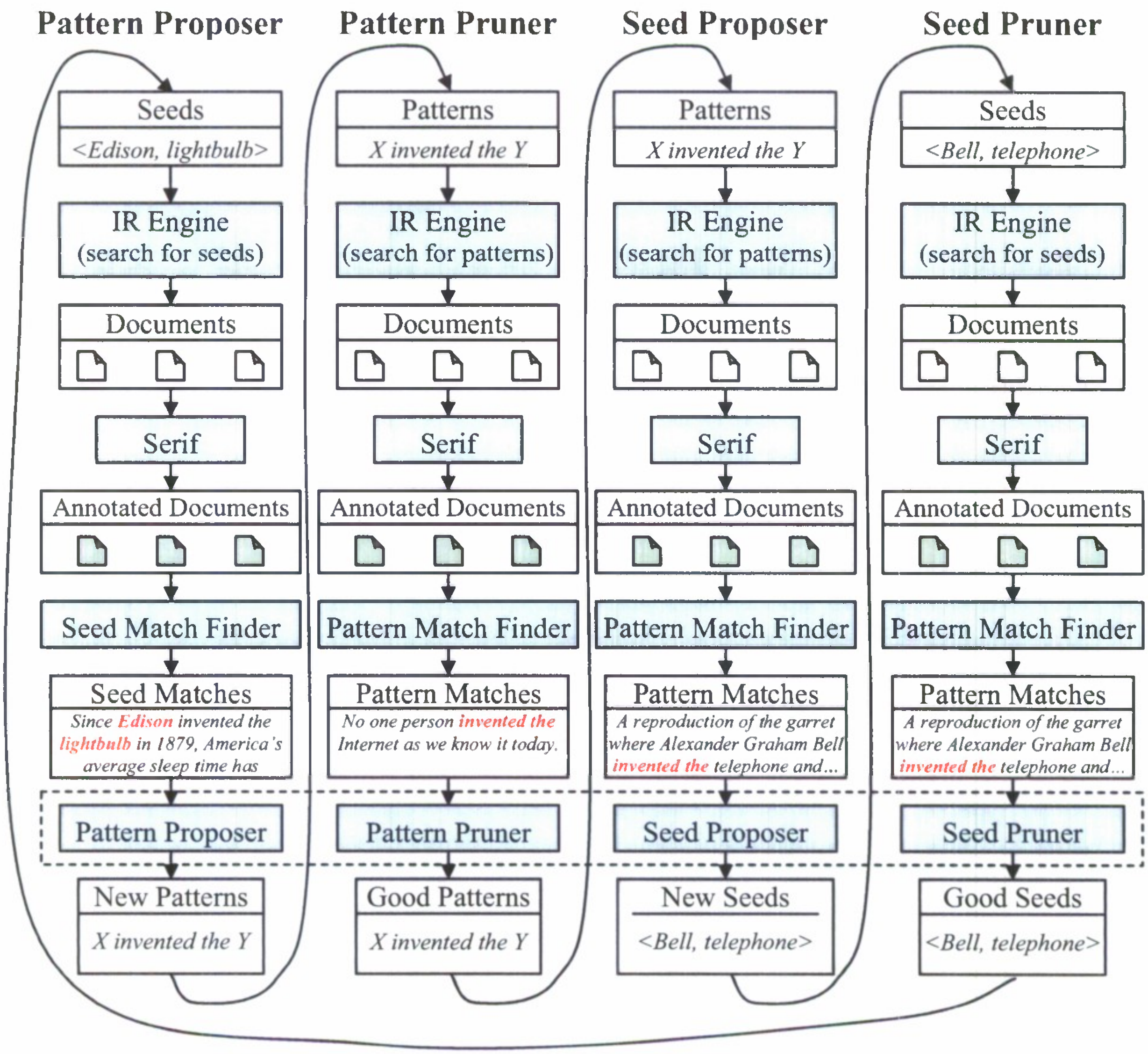

Figure 8-3: LearnIt System Workflow with Examples

The pattern proposer stage begins by using an information retrieval engine to search for the seed examples. It then uses Serif to automatically annotate those documents with syntactic and semantic information. The annotated documents are searched for sentences that contain the seeds. These sentences (known as "seed matches") are handed off to the pattern proposer, which uses them to suggest new patterns.

Next, the pattern pruner stage is used to evaluate these patterns and determine which ones are worth keeping. It begins by using the IR engine to search for documents that contain the patterns. This ensures that overly-general patterns can be detected and rejected (for example, the pattern "X's Y" could indicate any of several different relationships between X and Y, so it is therefore not a good pattern for indicating any single specific relationship). After processing these documents, it uses GALE Distillation software to identify sentences containing the proposed patterns. Finally, the pattern pruner assigns an approximate precision and recall score to each pattern, based on which seeds they did or did not find, and it uses a variable cutoff function to decide which patterns should remain active.

Having generated and vetted a new set of patterns, the system can now search for new seeds using the seed proposer stage. This stage begins by using the IR engine to search for the selected patterns. After processing the documents, it then uses the Distillation system to find sentences containing those patterns. Finally, any seed pairs that are found by that pattern are passed on to the seed pruner stage.

The seed pruner stage is responsible for determining the system's confidence in each of the seeds and deciding which seeds to keep. It begins by using the IR engine to find documents that contain the proposed seeds. It also uses search terms targeted to both the proposed seeds and the active patterns, in order to increase its likelihood of finding good matches. After running the documents through Serif, it searches them for all of the active patterns. Finally, the seed pruner calculates both a score and a confidence for each proposed seed, based on the set of patterns that matched them (and on the precision and recall scores of those patterns); and uses a variable cutoff function to decide which seeds should remain active.

Finally, the system returns to the pattern proposer stage, where it can now make use of the newly found seed examples to find more patterns. This cycle can be repeated for a fixed number of iterations, or until the LearnIt system is no longer able to find any new seeds or patterns. Table 8-5 shows several examples of the patterns that are found for the "business relationship" relation, when the system is run over Arabic text.

Y `authorized Colonel` ***X*** `... security`
`accompanying` ***Y*** `to the meeting,` ***X***
Y `... to meet his ...` ***X***
Y `and counterpart ...` ***X***

Table 8-5: Example string-based patterns for the "business relationship" relation (translated from Arabic). The ellipsis string ("...") can match zero to three words.

Improvements to Baseline System

We developed an initial version of this system in the summer and early fall of 2009. This section details further improvements made in the last quarter of 2009.

Learning Concepts

In our initial work on the LearnIt system, we focused on learning two-place predicates (relations), such as "X employs Y" or "X invented Y." During this quarter, we extended the system to also handle single-place predicates (concepts), such as "X is a person" or "X is an invention." In addition to being useful in their own right, we believe that these concepts can be leveraged to help improve performance of relation learning. For example, if we can learn a set of patterns that find inventions (even in contexts that do not mention the inventor), then we can develop a list of inventions, and then use that to help learn the "X invented Y" relationship.

To learn these unary concepts, we use patterns that indicate the entity's role in a proposition, as well as patterns that look for keywords near the entity on either side. Over-general patterns are even more of a problem with concepts than with binary relations, since they are only constrained by one slot. We therefore take into account the frequency that keywords appear in the corpus to alleviate this problem.

Eventually, as noted, we hope to jointly learn these unary concepts along with binary relations. For instance, when deciding whether X is the president of Y, we could restrict X to instances of the class 'politician'. We also hope to extend the system to support many-place predicates, such as "team X beat team Y in the Z game on date D with a score of A-to-B." When adding support for single-place predicates, we therefore chose a design that would facilitate this extension.

Constraints

Many of the relations that we might be interested in searching for are characterized by one or more implicit constraints. For example, for the "X is a parent of Y" relationship, we know that we should usually expect two X values for any given Y value. By allowing the user to document these constraints at the onset, we can make use of them to help guide the learning process. To test this idea, we added support for documenting two new constraints:

- `max-xs-per-y`: the maximum number of X values that we should expect to find for any given Y value.
- `max-ys-per-x`: the maximum number of Y values that we should expect to find for any given X value.

Whenever the LearnIt system detects that a pattern violates one of these constraints, it penalizes that pattern's precision score.

We found that these constraints improve the performance of the LearnIt system. However, some care must be taken in selecting appropriate constraint values. For example, although it initially seems like the relation "X was born on Y" should have exactly one Y for every X, further consideration reveals that it's possible to have one Y value that's a year ("1973"), while another is a date ("4/2"), and yet another is a day of

the week ("a thursday"). For this reason, we chose to use "soft constraints," which penalize patterns' scores, but do not automatically eliminate them from consideration.

Pattern and Seed Scoring

In our initial work on the LearnIt system, all patterns and scores were given binary scores (good or bad). However, this restricted our ability to make use of patterns that are indicative but not conclusive, such as "X authorized Y to..." (for the employment relationship). Additionally, the use of binary scores made it more difficult to combine information from multiple patterns to arrive at a decision for a given seed; or vice versa.

We therefore extended the system to assign scores to both seeds and patterns. Each seed is assigned a score from 0 (bad) to 1 (good), and a confidence from 0 (uncertain) to 1 (certain). The initial seed examples that are supplied by the user are given a score of 1.0 and a confidence of 1.0 (unless the user specifies otherwise). Each pattern is assigned an approximated precision score and recall score. The use of precision and recall scores for patterns is natural, since each pattern can be thought of as a type of complex search query. Furthermore, it allows us to use probabilities to combine information from different patterns in a principled way.

Pattern Scoring: Details

Pattern scoring is performed by the pattern pruner, which takes as input a list of sentences containing seed matches. It then calculates estimated precision and recall scores for each pattern by examining the set of seeds that matched that pattern:

$$\text{precision}(\textit{pattern}) = \sum_{\textit{seed} \in \textit{matches}(\textit{pattern})} \frac{\text{score}(\textit{seed}) * \text{confidence}(\textit{seed})}{\text{confidence}(\textit{seed})}$$

$$\text{recall}(\textit{pattern}) = \sum_{\textit{seed} \in \textit{database}} \frac{1(\textit{seed} \in \textit{matches}(\textit{pattern})) * \text{confidence}(\textit{seed})}{\text{confidence}(\textit{seed})}$$

The precision score is simply the confidence-weighted average of the scores of the matching seeds. The recall score is the confidence-weighted percent of the known-good seeds (from the database) that were successfully found by the pattern.

Usually, many of the seeds that are found by a pattern will be novel, and we will therefore have neither a score nor a confidence for them. For these seeds, we use a default score of zero, with a confidence of 1%. This choice of scores reflects the fact that most seed pairs are not good, but that we have very little information about these particular seeds. There are two cases in which we adjust this default score. First, if the seed matches an X from one database-seed with a Y from another database seed, then we use a score of zero and a confidence of 5%. This helps eliminate over-general patterns, such as "X's Y," which tend to match many combinations of Xs and Ys. Second, we use the equivalent names database (developed for the GALE Distillation program) to check if any similar seeds appear in our database. If so, we adjust the score of the novel seed in the direction of the similar seed, weighted by the similarity value that's given by the

equivalent names database. For example, if we have already found the seed <*Thomas Edison, lightbulb*>, and we see the novel seed <*T. Edison, lightbulb*>, then the equivalent names database will indicate that these are probably equivalent; and we will adjust the score of the novel seed toward the score of the known one.

Once we have calculated the precision and recall scores for a pattern, we then check for violations of the target relation's constraints (discussed above). In particular, we reduce the precision score by the percentage of seeds that had constraint violations. For example, if 34% of the seeds found by a pattern had constraint violations, then we would reduce the pattern's precision score by 34%.

Having scored the patterns, we apply a set of variable cutoffs to determine which patterns should be kept active for future searches. A separate cutoff function is applied for each of three metrics: precision, recall, and f-score (with α=0.9). Each cutoff function begins by sorting the patterns based on the cutoff score. The cutoff point is then determined using a rank-based curve, exemplified in Figure 8-4. In particular, the cutoff score begins with a fairly low value for the best pattern, but then gets progressively stricter for subsequent patterns. This ensures that we will usually get at least one or two patterns, but that we will only accept a large number of patterns if they are all of very high quality.

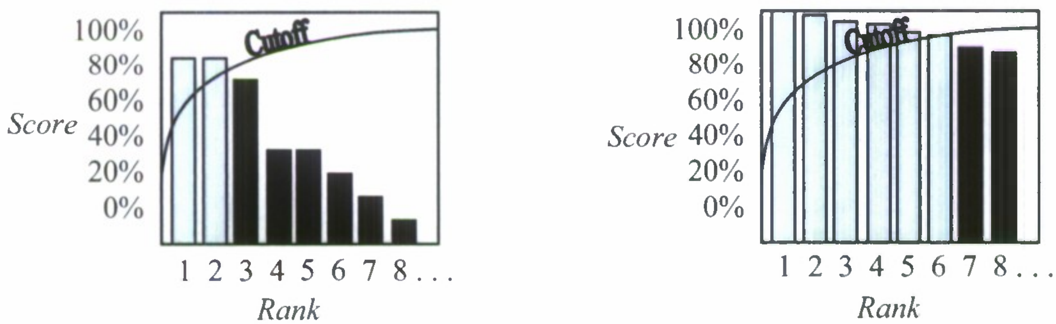

Figure 8-4: Rank-based cutoff curve. In the example on the left, the two highest-rated patterns fall above the cutoff curve, and are accepted; the remaining patterns are all rejected. In the example on the right, the six highest-rated patterns are accepted.

Seed Scoring: Details

Seed scoring is performed by the seed pruner, which takes as its input a list of sentences containing pattern matches. It begins by calculating the score of each seed, by finding the probability that the seed is good, given the set of patterns that match that seed. First, we use Bayes rule to decompose the conditional probability:

$$\begin{aligned} \text{score}(seed) &= \text{P}(seed^{+}|matches) \\ &= \frac{\text{P}(seed^{+}, matches)}{\text{P}(seed^{+}, matches) + \text{P}(seed^{-}, matches)} \end{aligned}$$

Key
$seed^{+}$: *seed* is good.
$seed^{-}$: *seed* is bad

Next, we apply the Naïve Bayes approximation, assuming that all patterns are independent:

$$P(\mathit{seed}^{+}, \mathit{matches}) \cong P(\mathit{seed}^{+}) \prod_{\substack{p \in \mathit{matching} \\ \mathit{patterns}}} P(\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{+}) \prod_{\substack{p \in \mathit{non\text{-}matching} \\ \mathit{patterns}}} P(\neg\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{+})$$

$$P(\mathit{seed}^{-}, \mathit{matches}) \cong P(\mathit{seed}^{-}) \prod_{\substack{p \in \mathit{matching} \\ \mathit{patterns}}} P(\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{-}) \prod_{\substack{p \in \mathit{non\text{-}matching} \\ \mathit{patterns}}} P(\neg\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{-})$$

To calculate $P(\mathit{seed}^{+})$, the prior on good seeds, we check what percentage of the seeds that were found by the pattern matches are already assumed to be good. We can calculate the probability of a pattern matching (or not matching) given that a seed is good (or bad) based on the precision and recall of the pattern.

$$\begin{aligned}
P(\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{+}) &= tp/(tp+fn) \\
P(\neg\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{+}) &= fn/(tp+fn) \\
P(\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{-}) &= fp/(fp+tn) \\
P(\neg\mathrm{match}(p,\mathit{seed})|\mathit{seed}^{-}) &= tn/(fp+tn)
\end{aligned}$$

Key
tn: % true negatives
fn: % false negatives
tp: % true positives
fn: % false positives

The confidence for a seed's score is calculated using the percentage of patterns that matched the seed: the more patterns that match the seed, the more confident we are in our score value.

Once the seeds have been scored, we apply a set of variable cutoffs, similar to the cutoffs used by the pattern pruner. We apply two cutoff functions: one for the score, and one for seed confidence.

Patterns

We implemented a new pattern type, the NestedPattern, which is used in cases where one of the seed's slots is subsumed by the other one. For example, this pattern type can be used to construct the following patterns:

- Y=[X's daughter] *parent(X,Y)*
- X=[the father of Y] *parent(X,Y)*
- X=[the Y year old …] *age(X,Y)*

When used in conjunction with Serif's coreference results, these patterns can find a large class of examples that were missed by the previous pattern types.

We also extended the regular-expression based KeywordPattern type with the ability to search for specific words within either of the slot fillers. For example, the following pattern searches for the pattern "X followed Y," but only where the X string contains the word "assassin":

- X=[…assassin…] followed Y *killed(X,Y)*

In addition, patterns can now be constructed that abstract over integers, floats, dates, and monetary amounts. For example, the "#INT#" expression in the following pattern can match any integer expression:

- Y=[... , where X has been since #INT#] *employs(X,Y)*

Miscellany

When using the information retrieval engine to find candidate documents for the two pruner stages, we now include queries that pair individual seeds with individual patterns. These queries increase the likelihood that we will find relevant documents; however, we can only afford to retrieve a few documents for each of these queries, because otherwise the number of documents we need to process would grow too large.

In addition, the search corpus will often contain multiple documents that are either exact copies of one another, or nearly identical. These duplications can cause a pattern or seed that appears in a duplicated document to be given a higher score than it deserves. We therefore updated the LearnIt system to detect and ignore duplicate sentences.

Evaluations

LearnIt Evaluation #1 (September)

The goal of these experiments was to determine how well LearnIt performed at a task similar to TAC slot-filling. The evaluation targeted thirteen relations, shown in the table below:

Person Relations	Organization Relations
Spouse	Top Members (ORG)
Siblings	Headquarters
Schools Attended	Founded By
Residences	Date Founded
Birth Place	
Death Place	
Birth Date	
Death Date	
Children	

Table 8-6: List of relations.

For these experiments, English Gigaword served as the large corpus. We began the learning process for each relation with 25 seeds, randomly chosen from an automatically extracted database of correct seeds for each relation type. We allowed only seeds which appeared a minimum of 10 times in the Gigaword corpus.

For each relation, we evaluated three different pattern sets generated by LearnIt:

1. **Standard**. The standard pattern set is the one produced by LearnIt after four iterations.

2. **Standard + Manual Pruning**. This pattern set is the result of allowing one minute of manual inspection per-relation (after all iterations are complete) to remove overly aggressive patterns. Typically, these patterns identified relations

in sentences that were correlated to correct answers, but did not actually include the relation. For instance, the string "X said Y" (where X and Y are names) often appears when X is a top member (leader) of Y, but the text does not provide justification for the top member relation.

3. **Recall-targeted**. The recall-targeted version of the system runs for between 6-16 iterations. At each iteration, it preserves more patterns than the standard version of the system. We allowed this system to run for up to 40 hours for each relation. The recall-targeted version of the system learned an order of magnitude more patterns than the initial version, but many of these patterns were very specific. For instance, "Y, widow of assassinated prime minister X" is a valid pattern for the spouse relation, but it is specific and will have very low recall. We ran a recall-targeted version of the system for seven of the test relations. As expected, the Recall-Targeted system learned many more patterns.

Evaluation Process

Rather than the expense of a manual evaluation, we chose an automatic evaluation so that more variations of the approach could be scored. That evaluation began by extracting gold seeds from world knowledge databases. We randomly chose 10% to be evaluated. To generate a key for the evaluation, we searched a test corpus (Gigaword 2005) for any sentences that contained both a subject and a corresponding correct answer. These were then considered the "gold sentences", i.e. the correct answers against which the automatic system would be evaluated.

Possible pitfalls of this approach were discussed in depth in the Y4Q2 quarterly report. To summarize, when examining instances found by only a single system, the human baseline was apparently penalized a net of 15% by the automatic scorer (compared to a manual scorer), while Standard LearnIt was given extra credit of 11% (before pruning) and 6% (after pruning). Therefore, the numbers reported below should be considered biased in favor of LearnIt. (For the subset of instances found by both systems, there is obviously no comparative bias, since each system received the same total score for that subset. This means the total bias is less than the percentages noted.)

Results

Table 8-7 shows the performance of all four systems, scored using a fully automatic process.

In all cases, the manual baseline had better recall than the Standard LearnIt system. In the majority of cases, however, the LearnIt patterns had better precision. (The manual system always has the better F-measure, partially because of the problems with artificially low recall across the board.)

After manual pruning, LearnIt's precision usually improved. (Recall sometimes went down, but this was largely due to the influence of the evaluation flaws described above; the patterns removed by the manual pruning usually produced only instances that were being incorrectly given credit in the first place.). In two cases, *Top Members* and *Death Place*, the automatic evaluation precision scores for the Standard+Pruning system were so much better than those of the manual baseline system that they represent real

improvement over the manual baseline, even after taking into consideration the automatic evaluation bias described above (which unfairly inflates the LearnIt scores).

For the recall-targeted system, which learned many more patterns, recall went up, in three of seven cases now beating the baseline, though precision tended to go down.

In the next table, places where a LearnIt system outperforms the manual baseline are highlighted in green. Places where either pruning or recall-targeting improves performance over the Standard LearnIt system are **underlined** (as expected, pruning improves precision and recall-targeting improves recall).

	Manual Baseline		Standard LearnIt		Standard LearnIt + Manual Pruning		Recall-Targeted LearnIt	
	P	R	P	R	P	R	P	R
Date Founded	0.9	0.11	1	0.06	1	0.06		
Founded By	0.87	0.07	0.42	0.04	1	0.01	0.58	0.19
Headquarters	0.6	0.18	0.61	0.11	0.62	0.11	0.58	0.11
Top Members	0.63	0.19	0.8	0.13	0.94	0.11	0.71	0.23
Children	0.89	0.18	0.79	0.02	0.79	0.02		
Birth Date	0.85	0.18	0.95	0.03	0.95	0.03		
Death Date	0.89	0.41	0.99	0.13	0.99	0.13		
Birth Place	0.75	0.02	0.55	0.01	0.78	0.01	0.49	0.04
Death Place	0.43	0.08	0.48	0.03	1	0.03	0.43	0.05
Residences	0.73	0.04	0.49	0.01	0.56	0.00	0.61	0.04
Schools Attended	0.83	0.12	0.97	0.03	0.97	0.03		
Siblings	0.46	0.07	0.67	0.01	0.67	0.01		
Spouse	0.82	0.17	0.83	0.03	0.84	0.03	0.64	0.11

Table 8-7: Automatic Evaluation Scores.

Further details of these experiments appeared in the Y4Q2 report.

Native Arabic SERIF vs. low-resource MT

Unsupervised techniques, such as those used by the LearnIt project, are a promising technique for finding relations in low-resource languages. However, we would also like to be able to leverage the many language processing tools, corpora, and lexicons that are available in English when working with these low-resource languages. Even though these resources are not directly applicable to the target language, we may be able to gain significant mileage by combining them with an even rudimentary machine learning system.

As a preliminary evaluation of this approach, we performed an experiment comparing the performance of the native Arabic SERIF system, which has been highly tuned to work on Arabic data (and requires supervised data for many component models), with the performance of running the English SERIF system over the output of a low-resource machine translation system. This low-resource MT system was trained using 1.5 million

words of parallel text. (In contrast, our state-of-the-art MT system is trained on 200 million words of parallel text). Unfortunately, it would be non-trivial to generate gold standard output for the MT system, because it would require aligning the relations in the source text with the relations in the target text. We therefore ran a simplified evaluation, where the task was to determine the set of relations that are expressed in a given sentence. This gives a rough approximation of the actual performance of the system. The results of this evaluation are shown in Table 8-8:

	Arabic SERIF			MT + English SERIF		
Relation	**P**	**R**	**F(1)**	**P**	**R**	**F(1)**
org-aff.employment	**80.6**	50.1	61.8	71.9	**56.8**	**63.5**
Part-whole.geographical	**68.7**	31.8	43.4	54.6	**45.0**	**49.3**
Gen-aff.citizen-resid.-relig.-ethnic.	**74.7**	40.2	**52.3**	53.7	**40.6**	46.2
Gen-aff.org-location	**64.8**	**30.4**	**41.3**	48.3	22.7	30.9
art.user-owner-inventor-manufact,	**72.9**	27.1	39.5	53.3	**40.0**	**45.7**
per-soc.business	**88.2**	**33.5**	**48.5**	67.3	12.8	21.5
Part-whole.subsidiary	**65.1**	46.5	**54.3**	48.2	**47.6**	47.9
per-soc.family	**89.5**	6.7	12.4	79.6	**49.0**	**60.7**
phys.located	**40.0**	7.2	12.2	22.3	**33.9**	**26.9**
org-aff.membership	**61.7**	31.9	**42.0**	39.7	**42.3**	41.0
Overall	**73.0**	33.9	46.3	54.3	**40.5**	**46.4**

Table 8-8: ACE Relation Scores. Comparison of the full Arabic SERIF system (run on the original Arabic text) to the English SERIF system (run on the output of a low-resource machine translation system), for all ACE-2005 relations with at least 100 test instances. The better score is indicated in **bold**.

These scores should be taken as an upper bound on the performance that the two systems would have, if they were required to find the actual extents of the two related entities. It should be noted that this metric may be more generous towards the MT system, where the noisy nature of MT might cause this system to correctly identify the presence of a relation, but be unable to determine the correct relation participants.

Overall, this experiment shows that the cross-language approach, of combining state-of-the-art English models with poor MT, shows promise. In general, the cross-language system has higher recall, though it also has universally lower precision. Taken as a whole, the resulting f-scores are quite close. It appears that the superior language tools, corpus sizes, and lexicons that are available in English roughly counterbalance the effects of poor machine translation. We expect that if this experiment were repeated in a language that is significantly more resource-poor than Arabic, then the cross-language system would most likely have a strong advantage.

LearnIt Evaluation #2 (December)

In December, in order to evaluate the performance of the LearnIt system more accurately using human assessors, we ran two new experiments, this time in both English and Arabic. The first experiment measures the precision of the patterns that are found by the LearnIt system; and the second experiment measures the recall of those patterns. The experiments were run on both English and Arabic text, using the following five relations:

- X employs Y.
- X killed Y.
- X is/was a spouse of Y.
- X was born at location Y.
- X was born on date Y.

For both experiments, the LearnIt system was initialized with 10 seeds, randomly chosen from a set of 30 seeds that were created by hand based on a brief web search including sites such as Wikipedia and `www.famousbirthdays.com`. It was then run for 20 iterations (or until it could find no new seeds or patterns).

For the English experiments, the LearnIt system was trained using the English Gigaword Corpus, which consists of 7 million documents containing a wide variety of genres, from newswire to fiction to web pages. For the Arabic experiments, LearnIt was trained using the Arabic Gigaword Corpus, which consists of approximately 300 thousand documents containing mostly newswire text.

Precision Experiment

In order to determine the precision of the patterns that were generated by the LearnIt system, we ran these patterns over new documents until we had collected 500 matching sentences, or until we had processed 50,000 documents. We then displayed the sentences that were found by LearnIt's patterns to human annotators, who were asked to judge whether they contained the desired relation between the indicated entities. The precision was then calculated as the percentage of patterns proposed by LearnIt which were judged to be correct by the human annotators. The results of this experiment are shown here:

Relation	English Precision	Arabic Precision
X employs Y	77%	86%
X killed Y	75%	19%
X has spouse Y	75%	77%
X has birth date Y	73%	50%
X has birth place Y	17%	20%

Table 8-9: Results from the precision experiment.

Recall Experiment

To measure the recall of the patterns that were generated by the LearnIt system, we selected 10 seeds that the LearnIt system had not encountered before. These seeds were selected randomly from the same pool of 30 seeds that were used to train the LearnIt system (ensuring that there was no overlap between these test seeds and the training seeds). We then used the information retrieval engine to search for any sentences containing these seed pairs; and had human annotators determine which of those sentences actually indicated the desired relation. Finally, we ran the patterns that were generated by LearnIt over the same set of documents, and calculated the percentage of the sentences that contain the desired relation that LearnIt was able to find. The results of the recall experiment are shown here:

Relation	English Recall	Arabic Recall
X has birth place Y	10.2%	2.0%
X killed Y	3.3%	0.5%
X has spouse Y	2.0%	0.5%
X has birth date Y	3.0%	1.0%
X employs Y	0.5%	0.5%

Table 8-10: Results from the recall experiment.

Discussion

As shown by the precision experiment, the LearnIt system is capable of learning robust patterns that can be used to find specific relations, starting with a very minimal set of training seeds. In fact, as is shown in Table 8-11, many of the seeds that were selected did not appear anywhere in the training corpus. For example, although we began with 10 seeds for the "X killed Y" relation, LearnIt was only able to make use of four of those seeds, since the remaining six seeds did not appear in the Gigaword corpus. Averaging across the relations, 24% of the English seeds and 36% of the Arabic seeds did not occur anywhere in the training corpus.

Relation	English	Arabic
X has birth place Y	9/10	10/10
X killed Y	4/10	4/10
X has spouse Y	10/10	7/10
X has birth date Y	7/10	3/10
X employs Y	8/10	8/10

Table 8-11: The number of training seeds appearing anywhere in training corpus.

The low recall numbers reflect the fact that there are a very wide variety of ways to express each of these relations. Many of these potential patterns merely imply a relationship, and further contextual or background information is necessary to determine that the relationship actually exits. If we wished to capture a larger portion of the relation occurrences, we could do so by running the LearnIt system for more iterations, generating a wider class of relation patterns. This would increase the overall recall of the patterns, at the expense of decreasing the overall precision. Another option to increase recall would be to initialize the system with a larger set of seeds, which would allow it to immediately take advantage of a larger portion of the unsupervised corpus to learn new relations.

The fact that the English system achieves higher performance than the Arabic system mainly reflects the fact that it was trained using an unsupervised corpus that is over twenty times larger than the corpus used for Arabic. In addition to the corpus size, the variety of genres in the corpus has a large impact on the overall performance. In particular, the presence of bibliographic articles in the English corpus is very useful for learning patterns for the relations we chose. In contrast, the Arabic corpus contains mostly newswire text, which is much less likely to contain any mentions of the selected

relations. For both of these reasons, the performance of the Arabic system would likely be improved by using a larger web-based corpus.

Table 8-12 gives a few examples demonstrating the output generated by the system. The first three samples show relations that LearnIt successfully found (true positives). The next two examples show incorrect relations that LearnIt found (false positives). The final two examples show correct relations that LearnIt failed to find (false negatives).

Example	Relation	Correct?	Found?
Martin Cooper <u>designs for</u> **Burberry** -- not the Prorsum line that gets all the press, but the more mass-appeal line that is more likely to be in a consumer's closet.	employs	Yes	Yes
``There have been a whole lot of linguists who have adopted it and seem -- for reasons that are incomprehensible to me -- to find it attractive," said **James McCawley**, a prominent <u>linguist at</u> **the University of Chicago**.	employs	Yes	Yes
Rafael Rodriguez, 24, was <u>gunned down</u> Jan. 2 by **a man riding a mountain bike** at 181st Street and Crotona Avenue in the start of a brief war between rival drug gangs.	killed	Yes	Yes
President Alberto Fujimori's lawyer said **Friday** that **Fujimori** <u>was born</u> in Peru, rejecting renewed allegations that he was born in Japan and his birth documents later altered.	has birth date	No	Yes
Visiting Chinese Vice-President Hu Jintao Tuesday met Japanese Prime **Minister Ryutaro** Hashimoto at his <u>official residence in</u> **Tokyo**.	has birth place	No	Yes
Aristotle, one of the great philosophers of **Ancient Greece**, who lived from 384 to 322 B.C., taught at the Lyceum from the age of 49 until his retirement at 62.	has birth place	Yes	No
In an exclusive interview with Le Figaro, French Interior Minister **Nicolas Sarkozy** listed the situation of Corsica, the Islamic organizations in **France** and the interior security as his priorities.	employs	Yes	No

Table 8-12: Sample Results for the English LearnIt system. The two related entities are shown in bold, and the text used by the pattern that found the relation is underlined.

Table 8-13 gives a few examples of the patterns that were learned by the LearnIt system. The *estimated precision and recall* columns show LearnIt's own internal estimation about how useful a pattern is. The *actual precision and recall* columns show the actual performance of the pattern, according to the experiments we ran. It should be noted that the recall scores for individual patterns will almost always be quite low – higher recall is generally gained by collecting a large number of patterns (the overall system recall is

approximately equal to the sum of the individual patterns' recall scores). In all cases, the patterns are restricted to find X and Y values of the appropriate type. For example, the birth date patterns (i) and (j) will only fire if X is a person, and Y is a time or date. Patterns with the form "proposition:P(role1=X, role2=Y)," such as examples (c), (d), (f), and (h), use Serif's semantic role labeling system to search for specific propositional relationships, regardless of how they are expressed syntactically.

	Relation	Pattern	Estimated Prec.	Estimated Rec.	Actual Prec.	Actual Rec.
a)	X killed Y	*X* was assassinated by *Y*	57%	13%	100%	1%
b)		*X*=[…, who was convicted of killing *Y*]	94%	8%	100%	0%
c)		proposition: murderer(ref=*X*, of=*Y*)	61%	17%	66%	2%
d)	X employs Y	proposition: designed(sub=*Y*, for=*X*)	62%	12%	100%	0%
e)		*Y* … *X* and Stella McCartney	94%	8%	0%	0%
f)		proposition: worked(at=*X*, with=*Y*)	60%	100%	10%	0%
g)		Y=[a linguistics professor at *X*]	64%	100%	8%	0%
h)	X has birth place Y	proposition: hometown(poss= *X*, of=*Y*)	64%	100%	20%	10%
i)		*X* resignation in *Y*	70%	0%	8%	0%
j)	X has birth date Y	*X* was born … *Y*	56%	17%	98.5%	3%
k)		*Y* (*X*	94%	3%	0%	0%

Table 8-13: Sample patterns learned by the English LearnIt system.

Patterns (a-d) are typical "good patterns," with fairly high precision scores. Pattern (c) is a good example of pattern overfitting: this pattern happened to be highly reliable for one of the seed examples, but does not generalize at all. Pattern (h) is an example of a pattern with unusually high recall (for an individual pattern). This probably reflects the fact that the "place of birth" relation does not get expressed very often; and when it does, it is often expressed by mentioning the word "hometown." Pattern (i) is an example of a pattern that gives a weak indication of a relation, but does not deserve the high estimated precision score that was assigned by the system. Pattern (k) is a good example of a pattern that is far too general. The LearnIt system found several documents that listed people, followed by parenthesized birth dates, and concluded that this was a reliable pattern. However, when it came to evaluation, this pattern did not score well.

Summary

LearnIt is a new system designed to use unsupervised techniques to learn specific relations and concepts. We have explored learning in both English (where parses and propositions are available) and Arabic (where we must rely on regular expressions and keywords). Initial results show promise, but further work is necessary to constrain the learning in a way that will provide better coverage without reducing precision. A first

step towards this is the incorporation of a minimal amount of human input at each iteration; work at BBN has already begun separately on this approach. We plan to continue to improve this system and hope to employ it in a wide variety of contexts going forward, including domain shift in English as well as in low-resource languages that have large corpora of (unannotated) text.